

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Podpora výuky předmětu Diskrétní metody a optimalizace
Diplomová práce

Autor: Bc. Aleš Krupička
Studijní obor: Aplikovaná informatika (ai2-k)

Vedoucí práce: RNDr. Andrea Ševčíková, Ph.D.

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové, 14. 8. 2020



Aleš Krupička

Poděkování:

Velmi děkuji vedoucí diplomové práce RNDr. Andree Ševčíkové, Ph.D. za metodické a odborné vedení práce, podnětné připomínky, lidský přístup a bezbřehou trpělivost při vzniku této práce.

Anotace

Multimediální aplikace jsou nástroje, které se často používají k podpoře vzdělávání. Mají za cíl studentům přiblížit a zprostředkovat koncepty a hůře pochopitelné procesy náročného učiva. Tato diplomová práce se zaměřuje na jeden takový nástroj, který byl v rámci práce vyvinut za účelem podpory výuky teorie grafů, zejména grafových algoritmů. Grafové algoritmy, vyučované v rámci předmětu Diskrétní metody a optimalizace, patří k obtížnějším učivu a pro studenty náročnějším pro pochopení. Teoretická část diplomové práce podrobněji seznamuje čtenáře s aplikovanými grafovými algoritmy. Stěžejní částí práce je webová aplikace, kterou popisuje praktická část. Ta usnadňuje výuku vyučujícím i studentům, vysvětluje pojmy a problémy teorie grafů v několika formách, které se vzájemně doplňují. Závěrečná část představuje testování aplikace.

Klíčová slova

Teorie grafů, diskrétní metody, graf, procházení grafu, komponenty grafu, kostra grafu, labyrint, eulerovský graf

Abstract

Title: Support for course the Discrete Methods and Optimization

Multimedia applications are often used to support the teaching process. The purpose of these tools is to clarify and mediate to students the concepts and less understandable aspects of schoolwork. The thesis focuses on one such tool; a web-based application developed within this work, to support the teaching of graph theory, particularly graph algorithms. Graph algorithms, taught in the Discrete Methods and Optimization course, belong to more difficult school subjects and are more demanding for students to understand. The theoretical part of the thesis acquaints the reader in more detail with the applied graph algorithms. The pivotal part of the thesis is the web-based application, which is described in the practical part. This application facilitates teaching for teachers as well as students, and outlines the concepts and problems of graph theory in several forms that complement each other. The final part presents the results of the application testing.

Keywords

Graph theory, discrete methods, graph, graph traversal, graph components, graph skeleton, labyrinth, Eulerian graph

Obsah

Úvod.....	13
1 Cíl práce	16
2 Metodika zpracování	17
3 Teoretický rámec diplomové práce	18
3.1 Grafové algoritmy na UHK.....	18
3.2 Základní grafové algoritmy	23
4 Aplikace.....	52
4.1 Návrh použitých nástrojů.....	52
4.2 Návrh databázové struktury.....	53
4.3 Implementace	58
4.4 Grafické rozhraní.....	59
4.5 Popis a ovládání aplikace	61
4.6 Ovládání administrátorské části.....	61
4.7 Ovládání studentské části	71
5 Testování aplikace	77
5.1 Podmínky testování.....	77
5.2 Průběh testování.....	77
6 Závěry a doporučení	89
7 Seznam použité literatury	90

Seznam obrázků

Obrázek 1 Rozmístění BTS společností T-Mobile v okrese Jeseník (zdroj [6])	14
Obrázek 2 Vyhledání trasy v navigaci prostřednictvím GPS (zdroj[7])	14
Obrázek 3 Vizualizace optimálních tras rozvozu od firmy SolverTech (zdroj [8]) ..	15
Obrázek 4 Ukázka Jarníkova algoritmu v programu GrAlg (zdroj [12])	19
Obrázek 5 Program Neuron (zdroj [15])	20
Obrázek 6 A-DIMA – pracovní matice (zdroj [14])	21
Obrázek 7 A-DIMA – okno algoritmu (zdroj [14])	21
Obrázek 8 Prohledávání do šířky (zdroj: vlastní zpracování)	24
Obrázek 9 Prohledávání do hloubky (zdroj: vlastní zpracování)	26
Obrázek 10 Jarníkův algoritmus (zdroj: vlastní zpracování)	28
Obrázek 11 Borůvkův algoritmus (zdroj: vlastní zpracování)	30
Obrázek 12 Kruskalův algoritmus (zdroj: vlastní zpracování)	31
Obrázek 13 Eulerovský tah (zdroj: vlastní zpracování)	32
Obrázek 14 Dijkstrův algoritmus (zdroj: vlastní zpracování)	34
Obrázek 15 Průběh Edmonds-Johnsonova algoritmu po pěti krocích (zdroj: vlastní zpracování)	36
Obrázek 16 Průběh Trémauxova algoritmu po pěti iteracích (zdroj: vlastní zpracování)	37
Obrázek 17 Zjišťování souvislosti grafu (zdroj: vlastní zpracování)	38
Obrázek 18 Zjišťování počtu komponent (zdroj: vlastní zpracování)	40
Obrázek 19 Vrcholy v jedné komponentě (zdroj: vlastní zpracování)	41
Obrázek 20 Testování, zda je hrana most (zdroj: vlastní zpracování)	42
Obrázek 21 Hledání délky nejkratší cesty (zdroj: vlastní zpracování)	43
Obrázek 22 Testování bipartity (zdroj: vlastní zpracování)	45
Obrázek 23 Vrchol na kružnici (zdroj: vlastní zpracování)	47
Obrázek 24 Hrana na kružnici (zdroj: vlastní zpracování)	48
Obrázek 25 Nalezení artikulací v grafu (zdroj vlastní zpracování)	49
Obrázek 26 Nalezení bloků v grafu (zdroj: vlastní zpracování)	51
Obrázek 27 Návrh úrovní aplikace (zdroj: vlastní zpracování)	52
Obrázek 28 Návrh struktury databáze, E-R diagram (zdroj: vlastní zpracování) ..	53
Obrázek 29 Úvodní stránka aplikace (zdroj: vlastní zpracování)	59

Obrázek 30	Prostor pro zobrazení grafu a práci s ním (zdroj: vlastní zpracování)	60
Obrázek 31	Ovládací prvky pro graf (zdroj: vlastní zpracování)	60
Obrázek 32	Přihlašovací formulář (zdroj: vlastní zpracování)	61
Obrázek 33	Rozcestník administrace (zdroj: vlastní zpracování)	61
Obrázek 34	Výpis algoritmů (zdroj: vlastní zpracování)	62
Obrázek 35	Editační formulář algoritmu (zdroj: vlastní zpracování)	63
Obrázek 36	Přehled grafů (zdroj: vlastní zpracování)	63
Obrázek 37	Editace grafu (zdroj: vlastní zpracování)	64
Obrázek 38	Nastavení manipulace s vrcholy a hranami (zdroj: vlastní zpracování)	64
Obrázek 39	Přehled ukázek (zdroj: vlastní zpracování)	65
Obrázek 40	Ovládání ukázky (zdroj: vlastní zpracování)	65
Obrázek 41	Výpis cvičení (zdroj: vlastní zpracování)	66
Obrázek 42	Úprava cvičení (zdroj: vlastní zpracování)	67
Obrázek 43	Výpis testových otázek (zdroj: vlastní zpracování)	67
Obrázek 44	Editace testové otázky (zdroj: vlastní zpracování)	68
Obrázek 45	Výpis pojmů ve slovníku (zdroj: vlastní zpracování)	68
Obrázek 46	Editace položky ve slovníku (zdroj: vlastní zpracování)	69
Obrázek 47	Výpis barev (zdroj: vlastní zpracování)	69
Obrázek 48	Editace Barev (zdroj: vlastní zpracování)	70
Obrázek 49	Přehled uživatelů (zdroj: vlastní zpracování)	70
Obrázek 50	Editace uživatelů (zdroj: vlastní zpracování)	71
Obrázek 51	Titulní stránka – přehled algoritmů (zdroj: vlastní zpracování)	71
Obrázek 52	Textový popis algoritmu (zdroj: vlastní zpracování)	72
Obrázek 53	Ukázka běhu algoritmu (zdroj: vlastní zpracování)	72
Obrázek 54	Videouázka algoritmu (zdroj: vlastní zpracování)	73
Obrázek 55	Cvičení algoritmu (zdroj: vlastní zpracování)	73
Obrázek 56	Vyřešení algoritmu na vlastním grafu (zdroj: vlastní zpracování)	74
Obrázek 57	Test pro konkrétní algoritmus (zdroj: vlastní zpracování)	74
Obrázek 58	Vysvětlení pojmu ve slovníku (zdroj: vlastní zpracování)	75
Obrázek 59	Úvodní obrazovka testu (zdroj: vlastní zpracování)	75
Obrázek 60	Editace testu studentem (zdroj: vlastní zpracování)	76
Obrázek 61	Zadání pro hledání minimální kostry (zdroj: vlastní zpracování)	80

Seznam tabulek

Tabulka 1 Porovnání aplikací GrAlg a teoriegrafu.cz (zdroj: vlastní zpracování, nástroj MS Excel)	22
Tabulka 2 Prohledávání do šířky (zdroj: vlastní zpracování, nástroj MS Excel).....	25
Tabulka 3 Prohledávání do hloubky (zdroj: vlastní zpracování, nástroj MS Excel)	27
Tabulka 4 Jarníkův algoritmus (zdroj: vlastní zpracování, nástroj MS Excel).....	28
Tabulka 5 Borůvkův algoritmus (zdroj: vlastní zpracování, nástroj MS Excel)	30
Tabulka 6 Eulerovský tah – průběh algoritmu (zdroj: vlastní zpracování, nástroj MS Excel)	33
Tabulka 7 Dijkstrův algoritmus (zdroj: vlastní zpracování, nástroj MS Excel)	35
Tabulka 8 Zjišťování souvislosti grafu (zdroj vlastní zpracování, nástroj MS Excel)	39
Tabulka 9 Zjišťování počtu komponent grafu (zdroj: vlastní zpracování, nástroj MS Excel)	40
Tabulka 10 Vrcholy v jedné komponentě (zdroj: vlastní zpracování, nástroj MS Excel)	41
Tabulka 11 Hledání nejkratší cesty mezi dvěma vrcholy (zdroj: vlastní zpracování, nástroj MS Excel)	44
Tabulka 12 Testování biparity – průběh algoritmu (zdroj: vlastní zpracování, nástroj MS Excel)	45
Tabulka 13 Vrchol na kružnici (zdroj: vlastní zpracování, nástroj MS Excel)	47
Tabulka 14 Nalezení artikulací v grafu (zdroj: vlastní zpracování, nástroj MS Excel)	50
Tabulka 15 Nalezení bloků v grafu (zdroj: vlastní zpracování, nástroj MS Excel) ...	51
Tabulka 16 Položky tabulky algoritmy (zdroj: vlastní zpracování, nástroj MS Excel)	54
Tabulka 17 Položky tabulky grafy (zdroj: vlastní zpracování, nástroj MS Excel).....	54
Tabulka 18 Položky tabulky ukázky (zdroj: vlastní zpracování, nástroj MS Excel).	55
Tabulka 19 Položky tabulky cvičení (zdroj: vlastní zpracování, nástroj MS Excel).	55
Tabulka 20 Položky tabulky barvy (zdroj: vlastní zpracování, nástroj MS Excel) ...	56
Tabulka 21 Položky tabulky slovník (zdroj: vlastní zpracování, nástroj MS Excel)	56
Tabulka 22 Položky tabulky otázky (zdroj: vlastní zpracování, nástroj MS Excel)..	57

Tabulka 23 Položky tabulky odpovědi (zdroj: vlastní zpracování, nástroj MS Excel)	
.....	57
Tabulka 24 Položky tabulky users (zdroj: vlastní zpracování, nástroj MS Excel)	58
Tabulka 25 Shrnutí výsledků testování (zdroj: vlastní zpracování, nástroj MS Excel)	
.....	88

Seznam grafů

Graf 1 Otázka 1, počty odpovědí (zdroj: vlastní zpracování, nástroj MS Excel).....	81
Graf 2 Otázka 1, procento správných a chybných odpovědí (zdroj: vlastní zpracování, nástroj MS Excel).....	81
Graf 3 Otázka 2, počty odpovědí (zdroj: vlastní zpracování, nástroj MS Excel).....	82
Graf 4 Otázka 2, procento správných a chybných odpovědí (zdroj: vlastní zpracování, nástroj MS Excel).....	82
Graf 5 Otázka 3, počty odpovědí (zdroj: vlastní zpracování, nástroj MS Excel).....	83
Graf 6 Otázka 3, procento správných a chybných odpovědí (zdroj: vlastní zpracování, nástroj MS Excel).....	83
Graf 7 Otázka 4, počty uvedených příkladů (zdroj: vlastní zpracování, nástroj MS Excel).....	84
Graf 8 Otázka 4, procento uvedených příkladů (zdroj: vlastní zpracování, nástroj MS Excel).....	84
Graf 9 Otázka 5, Jarníkův algoritmus, počty správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel).....	85
Graf 10 Otázka 5, Jarníkův algoritmus, procento správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel).....	85
Graf 11 Otázka 6, Kruskalův algoritmus, počty správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel).....	86
Graf 12 Otázka 6, Kruskalův algoritmus, procento správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel).....	86
Graf 13 Otázka 7, Borůvkův algoritmus, počty správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel).....	87
Graf 14 Otázka 7, Borůvkův algoritmus, procento správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel).....	87

Přehled použitého značení a pojmů

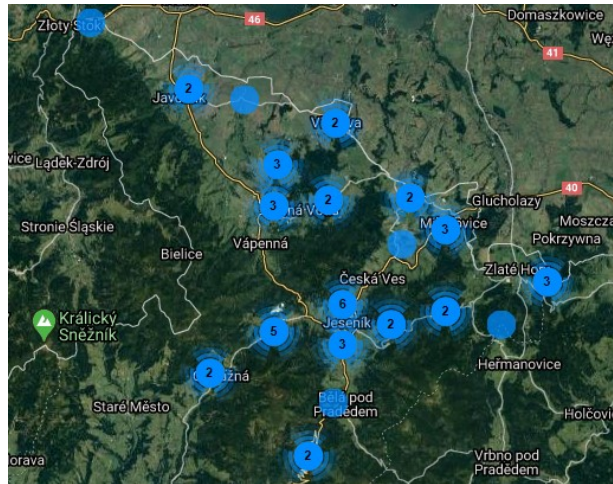
$G=(V, E)$	obyčejný graf, kde V je množina vrcholů a E je množina hran
$ V $	počet vrcholů
$ E $	počet hran
$-G$	doplňěk grafu
H	podgraf grafu
H	indukovaný podgraf, když pro každou dvojici vrcholů $\{v_1, v_2\} \in V(H)$: $(v_1, v_2) \in E(G)$, pak $(v_1, v_2) \in E(H)$
$G'(V, E')$	faktorový podgraf grafu G
A, B, C, \dots	prvky množiny vrcholů
a, b, c, \dots	prvky množiny hran
M	matice sousednosti
$\{A, B\}$	neorientovaná hrana mezi vrcholy A a B
AB	jiné zadání neorientované hrany
(A, B)	orientovaná hrana z vrcholu A do vrcholu B
$X \subseteq Y$	množina X je podmnožinou množiny Y
$X \cup Y$	sjednocení množin X a Y
$x \in X$	prvek x je prvkem množiny X
$deg(X)$	počet hran obsahujících vrchol X
$\delta(G)$	minimální stupeň grafu
$\Delta(G)$	maximální stupeň grafu
K_n	úplný graf na n vrcholech
$O(n)$	asymptotická časová složitost
<i>kostra grafu</i>	libovolný souvislý a acyklický faktor grafu
<i>uzavřený sled</i>	sled začínající a končící v tom samém vrcholu
<i>tah</i>	sled, v kterém se neopakuje žádná hrana
<i>uzavřený tah</i>	tah začínající a končící v tom samém vrcholu
<i>cesta</i>	tah, v kterém se neopakuje žádný vrchol
<i>souvislý graf</i>	graf, v kterém existuje sled (cesta) mezi libovolnou dvojicí vrcholů
<i>komponenta</i>	maximální souvislý podgraf
<i>most</i>	hrana e grafu G , že graf $G - e$ má více komponent než graf G
<i>artikulace</i>	vrchol v grafu G , že graf $G - v$ má více komponent než graf G

Úvod

Mnoho reálných problémů dnešního světa lze modelovat prostřednictvím matematických modelů. Matematické modelování se tak stává velmi významným vědeckým oborem, jehož uplatnění lze nalézt v mnoha dalších vědeckých oborech. Metody a algoritmy na bázi matematických modelů se následně denně uplatňují při řešení praktických úkolů.

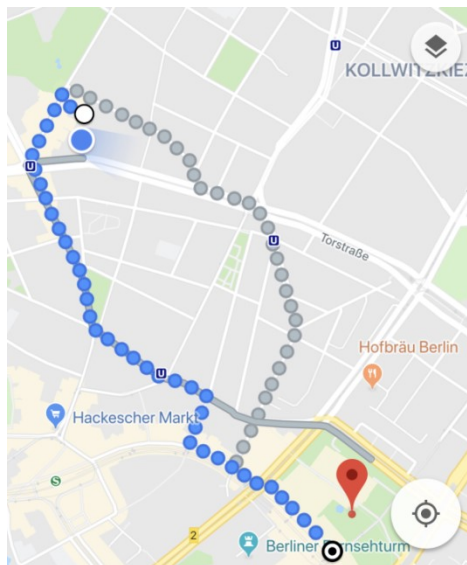
Významnou součástí matematického modelování je teorie grafů. Přestože počátky teorie grafů sahají až do 18. století, význam tohoto oboru roste zejména v posledních desetiletích ruku v ruce s rostoucím uplatněním v praxi, ale také s možností řešení prostřednictvím výpočetní techniky. Může jít o efektivní naplánování rozvodu jakékoliv sítě, např. dopravní, elektrické, počítačové apod. Další využití grafů je možné nalézt při často řešených problémech naplánování různě časově náročných úkonů, z nichž některé lze vykonávat paralelně. Typickým příkladem je stavba. Grafové algoritmy se také významně uplatňují při rozmístování určitých bodů v prostoru, např. vysílače, stanice záchranného systému, prodejny určitého typu, zdroje tepla ve větším prostoru apod., dále u logistických problémů a efektivním řízení provozu přepravy.[1], [2], [3], [4], [5]

Se stoupajícími nároky oborů, jako logistika, geografie, herní průmysl apod., lze předpokládat neutuchající využití modelování pomocí grafů a grafových algoritmů. Nahrávají tomu i zvětšující se možnosti výpočetní techniky. V současnosti jsou grafové algoritmy naprosto běžnou součástí života, aniž si to jejich uživatel uvědomuje. Takovým příkladem je rozmístění BTS (Base Transceiver Station) v krajině tak, aby bylo pokrytí mobilním signálem co nejvýhodnější. Grafové algoritmy jsou využity při zohledňování hustoty zalidnění, přírodních podmínek a dalších okolností ovlivňujících realizaci stavby BTS. Na obrázku 1 je možné vidět rozmístění BTS mobilního operátora T-Mobile v okrese Jeseník.[6]



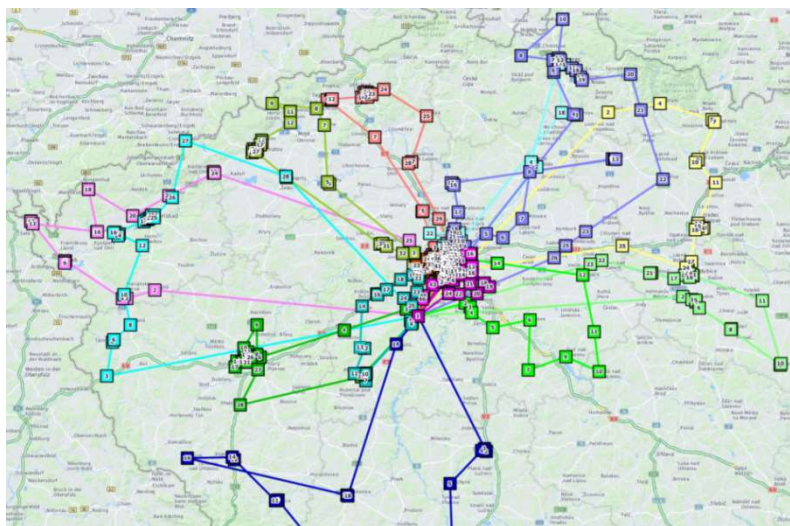
Obrázek 1 Rozmístění BTS společností T-Mobile v okrese Jeseník (zdroj [6])

Další oblastí, která je založená na grafových algoritmech, je ukládání tras v mapách, pro následné GPS vyhledání, viz obrázek 2. Pro uložení je nutné zohlednit typ komunikace, přírodní podmínky a další vlivy, pro vyhledání jsou zohledněny i aktuální faktory – typicky uzavírky silnic, dopravní nehody, překážky na silnici apod.[7]



Obrázek 2 Vyhledání trasy v navigaci prostřednictvím GPS (zdroj[7])

Bez nástrojů využívající grafové algoritmy se neobejde ani logistika. V této oblasti je nutné pružně reagovat na aktuální situaci (destinace aktuálních zakázek nákladní dopravy, zákazníci e-shopu nebo dopravní formy). Vstupní data se tedy mění denně a s nimi také výstup – optimální trasa vozů dodavatele, viz obrázek 3. [8]



Obrázek 3 Vizualizace optimálních tras rozvozu od firmy SolverTech (zdroj [8])

Ze zmíněného je patrné, že znalost práce s grafy, porozumění jejich zákonitostí a pochopení činnosti grafových algoritmů jsou pro praxi mnoha oborů velmi důležité. Z těchto důvodů jsou základy teorie grafů a především grafové algoritmy součástí výuky na Fakultě informatiky a managementu Univerzity Hradec Králové (dále FIM UHK) v rámci předmětů Diskrétní matematika (dále DIMA) a Diskrétní metody a optimalizace (dále DMO). Pro lepší porozumění a přiblížení průchodu grafových algoritmů používají vyučující různé multimediální aplikace a programy. Záměrem této práce je vytvoření výukové webové aplikace demonstrující základní grafové algoritmy, která už existující nástroje doplní a obohatí o dosud nevyužité funkce, čímž významně pomůže ulehčit výuku vyučujícím a studium studentům. Aplikace by se měla stát efektivní a účinnou pomůckou nejen pro studenty vysokých školy, ale pro každého, kdo se touto problematikou zabývá. Pro zvýšení efektivity výuky a individuálním potřebám uživatelů byla volena nejen textová, ale i audiovizuální forma. Postupným používáním a činnostmi administrátora by měla vzniknout rozsáhlá databáze grafů, souvisejících ukázek, cvičení, textů a testových otázek, která významnou měrou přispěje k pochopení problematiky.

Diplomová práce se skládá z teoretické a praktické části. Teoretická část se zabývá rešerší doposud existujících nástrojů na FIM a popisu základních grafových algoritmů použitých v aplikaci. V praktické části následuje popis tvorby samotné webové aplikace od jejího návrhu po implementaci řešení a rozbor její funkcionality. V závěru jsou popsány výsledky testování a používání včetně závěru a návrhů na rozšíření.

1 Cíl práce

Cílem této diplomové práce je vytvořit ucelenou webovou aplikaci, která nabídne studentům předmětu DMO (a také DIMA) nástroj pro pochopení principů probíraných algoritmů. Umožní také otestování dovedností a znalostí studenta.

Vzhledem k veřejné dostupnosti bude aplikace k dispozici každému, kdo se o problematiku grafových algoritmů z jakéhokoliv důvodu zajímá. Webová aplikace dále umožní administrátorům (vyučujícím) editovat obsah a rozšiřovat tak počáteční nabídku. Neklade si za cíl nahradit prezenční výuku. Jejím cílem je názornou formou demonstrovat průběh algoritmů. Kromě toho jsou algoritmy popsány textově a ve formě videa, což nabízí další alternativu usnadňující proces učení.

Tvorbě samotné aplikace předcházelo nastudování příslušných oblastí teorie grafů a teoretické zpracování grafových algoritmů. K tomu byla využita zejména literatura [2], [9], [10] a [11], existující diplomové a bakalářské práce na příbuzná témata, např. [12] a [13], popř. další zdroje.

Kromě samotné aplikace si diplomová práce klade za cíl vytvořit popis aplikace a návod k jejímu ovládní. Dalším dílčím cílem je sestavení databáze vhodných ukázkových příkladů a cvičení, které budou algoritmy demonstrovat a umožňují procvičení. Tyto příklady bude moci administrátor editovat, doplnit, mazat a postupně tak rozsah aplikace zvětšovat.

Závěrečnou částí práce je testování aplikace, za účelem demonstrace její přínosnosti. Pro testování byli záměrně zvoleni studenti, jejichž studijní předpoklady umožňují principy grafových algoritmů pochopit, ale zároveň nebyli před testováním s problematikou teorie grafů nijak obeznámeni – konkrétně půjde o studenty předmětu Seminář z programování v maturitním ročníku gymnázia, Praha 4, Budějovická 680. Testování bylo zpracováno pomocí MS Excel a vyhodnoceno porovnááním úspěšností jednotlivých testovacích skupin.

2 Metodika zpracování

Na základě stanovených cílů, po zvážení požadovaných vlastností aplikace a autorových znalostí a dovedností bylo rozhodnuto o následující metodice zpracování aplikace.

Aplikace je veřejná na doméně druhého řádu, která byla za tím účelem registrována. Současně s doménou byl zakoupen hostingový prostor, na nějž byly kladeny běžné nároky – zejména podpora jazyka PHP a podpora databáze MySQL. Následně byla databáze založena a také navržena její struktura.

Pro programování kostry aplikace a pro naprogramování manipulace s daty byl vzhledem ke zkušenostem autora zvolen programátorský jazyk PHP. K zobrazování grafů byla vybrána knihovna CytoScape, která pracuje na bázi JavaScriptu. Dále byla za účelem manipulace s daty na úrovni klienta, animací a také manipulaci dat na pozadí použita knihovna jQuery, také na bázi JavaScriptu. Výsledný kód byl vygenerován v jazyce HTML. Kromě toho byla využita knihovna mPDF pro generování souborů ve formátu *.pdf a knihovna tinyMCE pro formátování ve formulářových prvcích.

3 Teoretický rámec diplomové práce

Základem teoretického rámce této diplomové práce jsou základní grafové algoritmy probírané v rámci předmětu DMO a také DIMA na UHK.

3.1 Grafové algoritmy na UHK

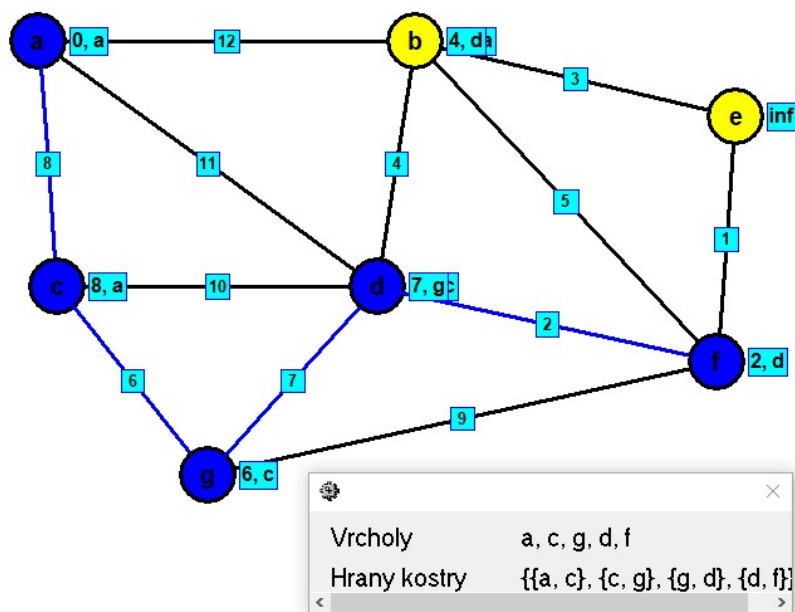
Pro grafové algoritmy vyučují se na FIM UHK byly v rámci diplomových a bakalářských prací v minulých letech na UHK pod vedením vyučujících vyvinuty různé interaktivní nástroje pro podporu výuky teorie grafů, jako GrAlg[12], A-DIMA [14], Neuron[15] a CheGra[16], webové nástroje GraPro [13], ProofVis [17] a DIMA – Teorie grafů – vizuální podpora [18] a mobilní aplikace GraphTheory [19]. Tyto nástroje jsou používány při objasňování základních konceptů teorie grafů, grafových algoritmů a důkazů matematických tvrzení z oblastí teorie grafů a také studentům slouží při domácím procvičování a přípravě ke zkouškám.

Mobilní aplikace Graph Theory a webová aplikace DIMA – Teorie grafů pomáhají studentům pochopit a procvičovat základní pojmy teorie grafů, webové aplikace GraPro a ProofVis jsou studentům nápomocné při výuce důkazů matematických vět z teorie grafů, aplikace CheGra nabízí studentům, hru inspirovanou jezdcovou procházkou, aplikací hamiltonovské kružnice. Nástroje GrAlg, A-DIMA a Neuron se týkají grafových algoritmů. Hlavním účelem GrAlg je nenáročné vytváření a modifikace grafů a vizualizace základních grafových algoritmů na grafech zadaných graficky. A-DIMA je aplikace, která demonstruje základní grafové algoritmy na grafech zadaných maticí sousedností. Studenti si v ní také mohou zkusit algoritmy a aplikace kontroluje správnost jednotlivých kroků studenta. Neuron se zabývá vizualizací grafových algoritmů pro barvení grafů. Těmto třem aplikacím týkajících se grafových algoritmů je věnovaná následující část, kde jsou podrobněji představeny:

3.1.1 GrAlg

GrAlg je program, který vznikl v roce 2010 jako diplomová práce na UHK. Jde o jediný EXE soubor bez nutnosti instalace. GrAlg umožňuje zadávat, editovat a mazat grafy a vyřešit základní grafové algoritmy, viz obrázek 4. Dále umožňuje grafy ukládat (vlastní formát *.grf) a znovu načítat, exportovat do BMP souboru a jako tabulku do MS Excel.

Výuková funkce programu spočívá zejména v postupném zpracování algoritmů a zobrazování jejich průběhu na pracovní ploše, zároveň jsou vypisovány relevantní údaje, například přidávané hrany. GrAlg umožňuje nastavovat rychlost algoritmu, průběh výpočtu zastavovat a opět spouštět. Toto nastavení mělo ale při testování na rychlost poměrně malý vliv (zvláště u grafů menšího rozsahu) a velmi rychle se objevil výsledek bez možnosti zásahu. Krokování algoritmů přímo možné není, je ale možné si animaci zastavit a opětovně spustit. Výhodou je anglická jazyková mutace.



Obrázek 4 Ukázka Jarníkova algoritmu v programu GrAlg (zdroj [12])

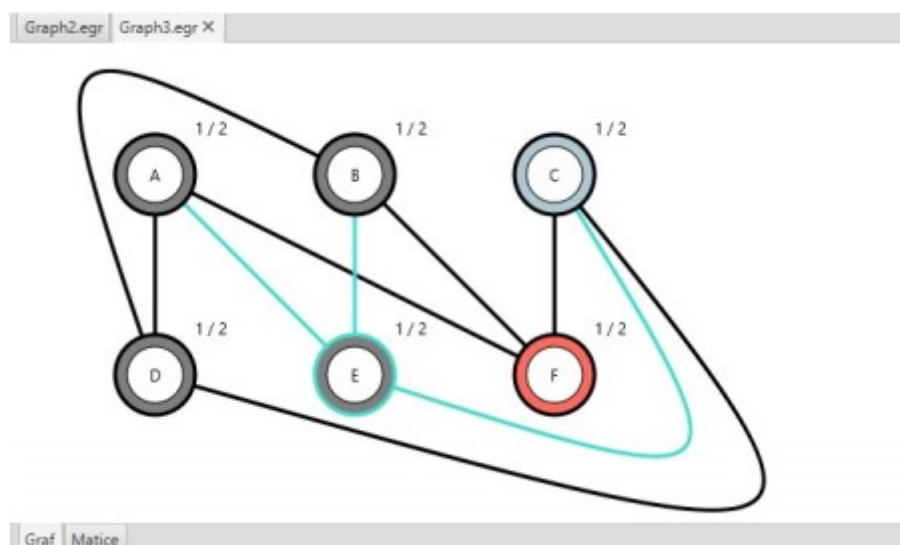
Ze subjektivního pohledu autora, jako uživatele, program GrAlg postrádá názornější demonstraci algoritmu, krokování algoritmu, databázi grafů a větší variabilitu při editaci grafu.

3.1.2 Neuron

Neuron je aplikace vytvořená jako součást diplomové práce Barvení grafů na UHK. Byl napsán v programovacím jazyku Java. Vizuálně přívětivým způsobem demonstuje tři algoritmy barvení grafů [20]:

- Sekvenční
- Nezávislé množiny
- Slepování vrcholů

Po nadefinování grafu a volbě algoritmu program aktivuje posuvnou lištu. Pohybem posuvníku lze algoritmus libovolně posouvat v obou směrech a zároveň sledovat změny v obarvení grafu. Současně se v dolní části okna objevuje textový výpis ke každému kroku. Kromě toho jde vrcholy grafu barevně označovat a měnit jejich velikost. Program Neuron je vizuálně vydařený a působí kompaktně. Splňuje účel, pro který byl stvořen. Tento program se zaměřuje pouze na specifickou oblast grafových algoritmů, proto porovnání s webovou aplikací z této práce není možné. [15]



Obrázek 5 Program Neuron (zdroj [15])

3.1.3 A-DIMA

A-DIMA byla vyvinutá na Katedře informatiky a kvantitativních metod FIM UHK jako podpora předmětu Diskrétní matematika v rámci bakalářské práce. Demonstruje grafové algoritmy na grafech zadaných maticí. Pracuje v několika oknech. Okno nazvané *Pracovní maticce* zobrazuje matici sousednosti, zde je také možné matici grafu zadávat a editovat, viz obrázek 6. [14]

	a	b	c	d	e
a	0	6	11	2	5
b	6	0	9	22	20
c	11	9	0	8	12
d	2	22	8	0	3
e	5	20	12	3	0

Obrázek 6 A-DIMA – pracovní matice (zdroj [14])

Po nadefinování grafu uživatel vybere požadovaný algoritmus a spustí jeho řešení. To se děje v *Okně algoritmu*, kde jsou také zobrazeny výsledky a průběh algoritmu. Vše je zobrazeno v textové podobě, program neobsahuje vizualizaci průběhu algoritmu na grafu, pouze na grafových strukturách, viz obrázek 7, což je pro studenty věnující se programování přínosné. Výhodou je velké množství algoritmů, které A-DIMA zvládá řešit, a také dvě jazykové mutace. Svým charakterem se tento program natolik odlišuje od webové aplikace z této práce, že srovnání není proveditelné.

Okno Algoritmu - Kruskalův Algoritmus	
Počet vrcholů:	5
Vrcholy grafu:	{a,}{b,}{c,}{d,}{e,}
Skóre grafu:	(4, 4, 4, 4, 4,)
Cena kostry:	19
Zpracované vrcholy:	a(2), b(2), c(2), d(2), e(2),
Zpracované hrany:	2ad, 3de, 5ae, 6ab, 8cd,
Hrany Kostry:	2ad, 3de, 6ab, 8cd,
Zde Zadejte hranu pro kontrolu:	<input type="text"/>
<input type="button" value="Ověřit"/>	
Kostra grafu už je kompletní	
<input type="button" value="Vyřešit najednou"/>	

Obrázek 7 A-DIMA – okno algoritmu (zdroj [14])

3.1.4 Porovnání aplikací

V následující tabulce 1 jsou přehledným způsobem porovnány nástroje GrAlg a teoriegrafu.cz.

Tabulka 1 Porovnání aplikací GrAlg a teoriegrafu.cz
(zdroj: vlastní zpracování, nástroj MS Excel)

Vlastnost	GrAlg	teoriegrafu.cz
1. Vkládání grafů	ano	ano
2. Mazání vrcholů	ano	ano
3. Mazání hran	ano	ano
4. Editace umístění vrcholů	ano	ano
5. Editace hodnocení hran	ano	ano
6. Editace popisu vrcholů	ano	ano
7. Editace barev vrcholů	ano	ano
8. Editace barev hran	ano	ano
9. Editace orientace hran	ne	ano
10. Změna měřítka grafu	ano	ano
11. Uložení grafu do souboru	ano	ano
12. Uložení grafu do databáze	ne	ano, admin
13. Export grafu do obrázku	ano	ano
14. Komentované ukázky	ne	ano
15. Procvičování	ne	ano
16. Přístupnost na internetu	ne	ano
17. Nutnost instalace	ne	ne
18. Slovní popis algoritmu	ne	ano
19. Slovník pojmů	ne	ano
20. Test znalostí	ne	ano
21. Možnost krokování ukázek	ne ¹	ano
22. Videoukázky algoritmů	ne	ano
23. Export testu do pdf	ne	ano, admin
24. Podgraf	náhled	ne
25. Doplnkový graf	náhled	ne
26. Skóre grafu	ano	ano
27. Anglická jazyková mutace	ano	ne

Stávající nástroje pro podporu výuky grafových algoritmů na UHK jsou jistě přínosné. Doposud ale chyběl nástroj disponující širšími možnostmi využití. Mezi nejdůležitější patří všeobecná dostupnost na internetu, databáze příkladů, celková editovatelnost a více vzájemně se podporujících forem výuky (interaktivní příklady, videoukázky, testy, textová podoba).

¹ Je možné algoritmus zastavit a opětovně spustit

3.2 Základní grafové algoritmy

Základní pojmy aplikované při popisu základních grafových algoritmů probíraných v rámci předmětu DMO v následující části jsou uvedeny v Přehledu použitého značení a pojmů. Přesné definice lze najít v [2], [9].

U každého algoritmu je uvedena asymptotická časová složitost. Asymptotická časová složitost algoritmu (dále jen časová složitost) A je řád růstu funkce $f(n)$, která charakterizuje počet elementárních operací algoritmu A při zpracování dat o rozsahu n . Nejpoužívanější třída asymptotické složitosti algoritmů $O(g(n))$ je definována: Necht' g je funkce. Pak definujeme množinu funkcí $O(g(n))$ takto: [21], [22]

$$O(g(n)) = \{f \mid \exists c > 0; n_0 \in \mathbb{N}: \forall n > n_0: |f(n)| \leq |c \cdot g(n)|\}$$

3.2.1 Prohledávání do šířky (BFS)

Algoritmus **prohledávání do šířky** (breadth-first search, BFS) prohledává všechny vrcholy a hrany příslušné komponenty souvislosti. Vrcholy jsou prohledávány po jednotlivých úrovních. Jsou tedy vždy prohledány všechny sousední, dosud neprohledané vrcholy aktuálního vrcholu. Při ukládání vrcholů je použita datová struktura FIFO, tedy fronta, jinak řečeno roura.[2],[23]

Následující formulace algoritmu je dle[2]:

Kroky algoritmu

VSTUP: souvislý graf $G(V, E)$

ÚKOL: systematicky zpracovat všechny vrcholy a hrany

POMOCNÁ PROMĚNNÁ: Fronta (FIFO) obsahující vrcholy, které již byly zpracované, a je ještě možné z nich pokračovat dál.

KROK 1: Vrchol $A \in V$ označ jako počáteční vrchol.

KROK 2: Všem vrcholům grafu nastav příznak neprohledán.

KROK 3: Počátečnímu vrcholu A nastav příznak prohledán.

KROK 4: Vlož počáteční vrchol do fronty FIFO.

KROK 5: Dokud není fronta FIFO prázdná, opakuj:

- a. První vrchol fronty FIFO označ příznakem prohledán a odeber ho z fronty FIFO.
- b. Všechny vrcholy, do nichž vede hrana z právě odebraného vrcholu, a mají příznak neprohledán, přesuň na konec fronty.

Složitost algoritmu

Složitost algoritmu prohledávání do šířky je $O(|V|+|E|)$. Každý vrchol a každá hrana je prohledána právě jednou.[2]

Z FIFO se odebírá vždy prvek, který se ve frontě nachází nejdéle [10].

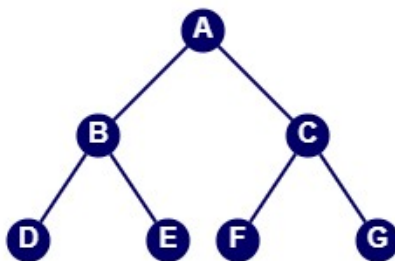
Využití algoritmu

Algoritmus prohledávání se využívá při řešení dalších problémů teorie grafů, například [24]:

- zjišťování souvislosti grafu,
- zjišťování počtu komponent,
- testování, zda je hrana most,
- testování, zda dva vrcholy leží v téže komponentě,
- zjišťování nejkratší cesty neohodnoceného grafu,
- testování bipartity grafu
- testování, zda vrchol leží na kružnici

Ukázka algoritmu

Na obrázku 8 je znázorněn graf, na němž je ukázka algoritmu demonstrována.



Obrázek 8 Prohledávání do šířky (zdroj: vlastní zpracování)

V tabulce je znázorněn průběh algoritmu prohledávání do šířky. Start je ve vrcholu A.

Tabulka 2 Prohledávání do šířky
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Fronta	Navštívené	Nenavštívené	Přidané hrany
1.	A		ABCDEFGG	
2.	BC	A	BCDEFG	
3.	CDE	AB	CDEFG	AB
4.	DEFG	ABC	DEFG	AC
5.	EFG	ABCD	EFG	BD
6.	FG	ABCDE	FG	BE
7.	G	ABCDEF	G	CF
8.		ABCDEFGG		CG

3.2.2 Prohledávání do hloubky (DFS)

Algoritmus **prohledávání do hloubky** (deep-first search, DFS) prohledává všechny vrcholy a hrany příslušné komponenty souvislosti. Pořadí prohledávaných vrcholů je určeno tak, že je ihned opouštěna aktuální vrstva a následující prohledávaný vrchol je ve vrstvě nižší. Návrat nastává až v momentě, kdy už není jiná možnost (v případě stromů tehdy, je-li prohledávaným vrcholem list). Při programové realizaci prohledávání do hloubky je vhodné využít rekurse. Při ukládání vrcholů je použita datová struktura LIFO, tedy zásobník. [25], [23] Následující formulace algoritmu je dle [2]:

Kroky algoritmu

VSTUP: souvislý graf $G(V, E)$

ÚKOL: systematicky zpracovat všechny vrcholy a hrany

POMOCNÁ PROMĚNNÁ: Zásobník (LIFO) obsahující vrcholy, které již byly zpracované, a je ještě možné z nich pokračovat dál

KROK 1: Vrchol $A \in V$ označ jako počáteční vrchol.

KROK 2: Všem vrcholům grafu nastav příznak neprohledán.

KROK 3: Počátečnímu vrcholu A nastav příznak prohledán.

KROK 4: Vlož počáteční vrchol A do zásobníku LIFO.

KROK 5: Dokud není zásobník LIFO prázdný, opakuj:

- První vrchol zásobníku LIFO označ příznakem prohledán a odeber ho ze zásobníku LIFO.
- Všechny vrcholy, do nichž vede hrana z právě odebraného vrcholu a mají příznak neprohledán, přesuň na začátek zásobníku LIFO.

Složitost algoritmu

Složitost algoritmu prohledávání do šířky je $O(|V|+|E|)$. Každý vrchol a každá hrana je prohledána právě jednou. [2]

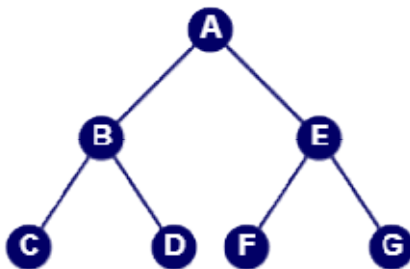
Využití algoritmu

Algoritmus prohledávání do hloubky se využívá při řešení dalších problémů teorie grafů, například[24]:

- zjišťování souvislosti grafu,
- zjišťování počtu komponent,
- testování, zda je hrana most,
- testování, zda dva vrcholy leží v téže komponentě,
- zjišťování nejkratší cesty neohodnoceného grafu,
- testování bipartity grafu,
- testování, zda vrchol leží na kružnici,
- zjišťování počtu artikulací

Ukázka algoritmu

Na obrázku 9 je znázorněn graf, na němž je ukázka algoritmu demonstrována.



Obrázek 9 Prohledávání do hloubky (zdroj: vlastní zpracování)

V tabulce 3 je znázorněn průběh algoritmu prohledávání do šířky. Start je ve vrcholu A.

Tabulka 3 Prohledávání do hloubky
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Zásobník	Navštívené	Nenavštívené	Přidané hrany
1.	A		ABCDEFG	
2.	AB	A	BCDEFG	
3.	ABC	AB	CDEFG	AB
4.	ABD	ABC	DEFG	BC
5.	AE	ABCD	EFG	BD
6.	AEF	ABCDE	FG	AC
7.	AEG	ABCDEF	G	EF
8.		ABCDEFG		EG

3.2.3 Jarníkův algoritmus

Jarníkův algoritmus je první ze tří zde uvedených grafových algoritmů, které jsou určeny pro nalezení minimální kostry grafu. V anglicky psané literatuře je uváděn jako Primův algoritmus. Princip algoritmu spočívá v postupném přidávání hran s co nejnižším ohodnocením, které netvoří kružnici. [26], [9],[27] Následující formulace algoritmu je dle[2]:

Kroky algoritmu

VSTUP: souvislý graf $G(V, E)$

ÚKOL: najít minimální kostru grafu

POMOCNÁ PROMĚNNÁ: seznam možných hran (reprezentovaný polem) a jejich ohodnocení.

KROK 1: Vrchol $A \in V$ označ jako aktuální vrchol a vlož ho do kostry.

KROK 2: Incidenční hrany vrcholu A vlož do seznamu možných hran.

KROK 3: Dokud nejsou v kostře všechny vrcholy, opakuj:

- a. Ze seznamu možných hran najdi tu s nejmenším ohodnocením.
- b. Vrchol této hrany, který dosud nebyl v kostře, do kostry přidej.
- c. Odeber tuto hranu ze seznamu možných hran.
- d. Incidenční hrany právě přidaného vrcholu, jejichž druhý vrchol leží v kostře, odeber ze seznamu možných hran.
- e. Incidenční hrany právě přidaného vrcholu, jejichž druhý vrchol neleží v kostře, přidej do seznamu možných hran.

Složitost algoritmu

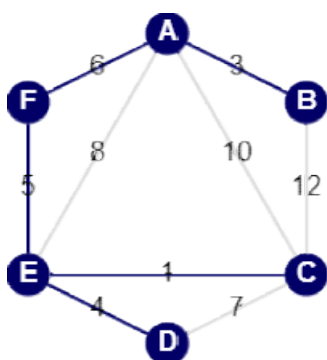
Složitost Jarníkova algoritmu je $O(|E|.log_2|V|)$ [2].

Využití algoritmu

Minimální kostra je modelem pro mnoho reálných situací, např. stavba dopravní sítě, elektrického vedení apod.[10]

Ukázka algoritmu

Pro ukázkou Jarníkova algoritmu byl zvolen graf na obrázku 10. Je na něm zvýrazněna i výsledná minimální kostra.



Obrázek 10 Jarníkův algoritmus (zdroj: vlastní zpracování)

Průběh Jarníkova algoritmu je patrný z tabulky 4. Prvním vrcholem je A.

Tabulka 4 Jarníkův algoritmus
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Přidané vrcholy	Aktuální hrana	Přidané hrany	Nepřidané vrcholy
1.	A			BCDEF
2.	AB	AB (3)	AB	CDEF
3.	ABF	AF (6)	AB, AF	CDE
4.	ABFE	EF (5)	AB, AF, EF	CD
5.	ABCEF	CE (1)	AB, AF, EF, CE	D
6.	ABCDEF	DE (4)	AB, AF, EF, CE, DE	

3.2.4 Borůvkův algoritmus

Borůvkův algoritmus nalezne minimální kostru prostřednictvím skládání komponent. Na začátku algoritmu jsou všechny vrcholy považovány za komponenty. Komponenty jsou procházeny cyklem a spojovány hranou s nejnižším ohodnocením, dokud nezůstane jediná komponenta – minimální kostra.[10], [26],[27]

Následující formulace algoritmu je dle[2]:

Kroky algoritmu

VSTUP: souvislý graf $G(V, E)$

ÚKOL: najít minimální kostru grafu

POMOCNÉ PROMĚNNÉ:

- seznam možných hran každého stromu (reprezentovaný polem) a jejich ohodnocení,
- počet stromů.

KROK 1: Každý vrchol grafu označ jako samostatný strom.

KROK 2: Dokud je počet stromů větší než 1, opakuj KROK 3.

KROK 3: Pro každý strom opakuj:

- a. Najdi hranu s minimálním ohodnocením z těch hran, jejichž jeden vrchol leží v daném stromu a druhý ne.
- b. Přiřaď tuto hranu do kostry.
- c. Přidaný strom sluč s původním stromem.
- d. Dekrementuj počet stromů.

Složitost algoritmu

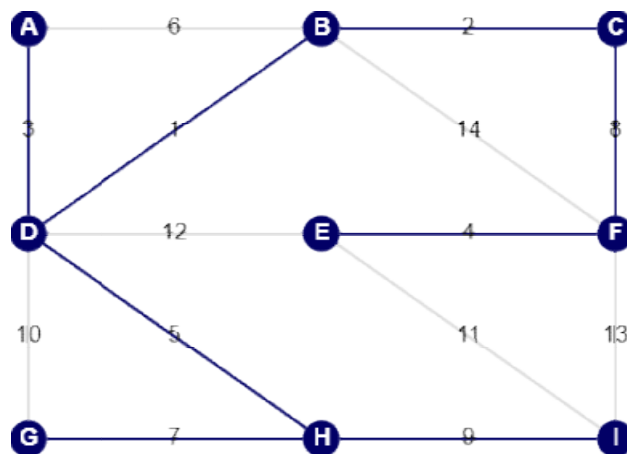
Složitost Borůvkova algoritmu je $O(|E| \cdot \log_2 |V|)$. Proběhne $\log_2 |V|$ fází a každá fáze je lineární. [2]

Využití algoritmu

„O. Borůvka algoritmus hledání nejlevnější kostry objevil jako řešení konkrétní praktické úlohy elektrifikace území. V této úloze obce tvoří vrcholy grafu, elektrická spojení hrany a ohodnocení hran reprezentuje cenu za natažení vedení mezi obcemi. Úkolem bylo najít takové spojení, aby všechny obce byly připojeny do elektrické sítě a přitom cena spojení byla minimální.“ [1]

Ukázka algoritmu

Graf zvolený pro ukázkou Borůvkova algoritmu a výsledná minimální kostra je na obrázku 11.



Obrázek 11 Borůvkův algoritmus (zdroj: vlastní zpracování)

Tabulka 5 znázorňuje postupné slučování stromů během Borůvkova algoritmu.

Tabulka 5 Borůvkův algoritmus
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Komponenty	Vybrané hrany
1.	A, B, C, D, E, F, G, H	AD, BD, BC, EF, GH, DH, IH
2.	ABCDGH, EF	CF
3.	ABCDEFGH	

3.2.5 Kruskalův algoritmus

Kruskalův algoritmus je také určen k nalezení minimální kostry v grafu. Nejdříve seřadí hrany vzestupně podle jejich ohodnocení. Ve druhé fázi postupně hrany přidává do kostry, ale pouze ty, které nevytvoří kružnici. [2], [26], [27] Následující formulace algoritmu je dle[2]:

Kroky algoritmu

VSTUP: souvislý graf $G(V, E)$

ÚKOL: najít minimální kostru grafu

POMOCNÁ PROMĚNNÁ: seznam možných hran (reprezentovaný polem) a jejich ohodnocení.

KROK 1: Seřaď seznam hran grafu vzestupně podle hodnocení.

KROK 2: Dokud nejsou v kostře všechny vrcholy, opakuj:

- a. Ze seznamu hran najdi takovou hranu s nejmenším ohodnocením, jejíž jeden vrchol oba vrcholy neleží v kostře
- b. Přidej hranu a její vrcholy do kostry.

Složitost algoritmu

Složitost Kruskalova algoritmu je $O(|E| \cdot \log_2 |E|)$. [2]

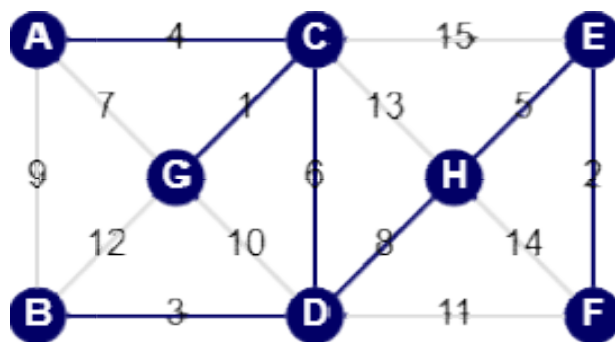
Využití algoritmu

Stejně jako u předchozích dvou algoritmů je využití minimální kostry ve všech případech budování sítě [10].

Ukázka algoritmu

Seřazené hrany: CG(1), EF(2), BD(3), AC(4), EH(5), CD(6), AG(7), DH(8), AB(9), DG(10), DF(11), BG(12), CH(13), FH(14), CE(15)

Přidávání hran: CG(1), EF(2), BD(3), AC(4), EH(5), CD(6), DH(8). Ostatní hrany by přidáním vytvořily kružnici. Výsledná kostra je na obrázku 12.



Obrázek 12 Kruskalův algoritmus (zdroj: vlastní zpracování)

3.2.6 Eulerovský tah

Algoritmus nalezne v grafu eulerovský tah. Jako první krok nalezne algoritmus libovolnou kružnici ze startovacího vrcholu. Poté kružnici prochází, a pokud u aktuálního vrcholu existuje neprobádaná hrana, vydá se po této hraně, dokud se opět nedostane do výchozího vrcholu. [11]

Následující formulace algoritmu je dle [11]:

Kroky algoritmu

VSTUP: souvislý graf $G(V, E)$, který má všechny vrcholy sudého stupně (uzavřený eulerovský tah) nebo právě dva vrcholy lichého stupně (otevřený eulerovský tah)

ÚKOL: najít eulerovský tah

KROK 1: Vyber libovolný vrchol (pokud jsou v grafu dva vrcholy lichého stupně, musí to být jeden z nich) a najdi kružnici T , která jím prochází.

KROK 1: Tuto kružnici vlož do eulerovského tahu.

KROK 2: Dokud nejsou v eulerovském tahu všechny hrany, opakuj:

- a. Procházej dosavadní eulerovský tah a najdi první vrchol, který obsahuje nepoužité hrany.
- b. Z tohoto vrcholu najdi kružnici zpět do původního bodu.
- c. Tuto kružnici přidej do eulerovského tahu.

V případě dvou vrcholů lichého stupně je algoritmus modifikován tak, že je nutné začít v jednom z těchto vrcholů. Následně nehledáme kružnici, ale cestu do druhého z vrcholů lichého stupně.

Složitost algoritmu

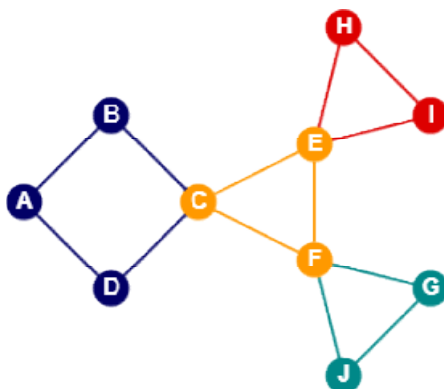
Složitost nalezení eulerovského tahu je $O(|E| \cdot \log_2 |E|)$. Počet procházených hran je $|E|$, počet návratů $\log_2 |E|$. [11]

Využití algoritmu

Využití eulerovských tahů je například ve stavových grafech, kde jednotlivé vrcholy představují stavy automatu a hrany přechody mezi stavy. [28]

Ukázka algoritmu

Na obrázku 13 je graf, který byl zvolen pro ukázkou nalezení eulerovského tahu.



Obrázek 13 Eulerovský tah (zdroj: vlastní zpracování)

Startovací vrchol A. Startovací vrchol je A. Tabulka 6 zachycuje postupně se zvětšující eulerovský tah. Na obrázku 13 jsou jednotlivé kroky barevně odlišeny.

Tabulka 6 Eulerovský tah – průběh algoritmu
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Kružnice	Vkládaná od-bočka	Vrcholy s nepoužitými hranami
1.	ABCD A		C
2.	ABCEFCDA	CEFC	EF
3.	ABCEHIEFCDA	EHIE	F
4.	ABCEHIEFGJFCDA	FGJF	

3.2.7 Dijkstrův algoritmus

Dijkstrův algoritmus nalezne nejkratší cesty od zadaného vrcholu ke všem ostatním vrcholům.[29]

Následující formulace algoritmu je dle[29]:

Kroky algoritmu

VSTUP: souvislý graf $G(V, E)$, s nezáporným ohodnocením w , počáteční vrchol A , prázdná množina N prohledaných vrcholů.

ÚKOL: nalezení nejmenší vzdálenosti od počátečního vrcholu ke všem ostatním vrcholům

POMOCNÉ PROMĚNNÉ:

- Množina známých vrcholů k prohledání,
- Množina již uzavřených vrcholů.

KROK 1: Nastav A jako aktuální vrchol.

KROK 2: Nastav počáteční vzdálenosti od vrcholu A :

- $d[A] = 0$
- $d[V_j] = w(A, V_j)$ pro všechny sousední vrcholy vrcholu A
- $d[V_i] = \infty$ pro všechny ostatní vrcholy

KROK 3: Dokud nejsou v eulerovském tahu všechny hrany, opakuj:

- Projdi všechny sousední vrcholy aktuálního vrcholu a každému z nich nastav vzdálenost od vrcholu A jako minimum:
 - aktuální vzdálenosti od bodu A (nedojde ke změně).
 - součtu vzdálenosti aktuálního bodu od bodu A a ohodnocení spojovací hrany (dojde ke změně).

- b. Uzavři aktuální vrchol.
- c. Jako nový aktuální vrchol zvol ten s nejmenší vzdáleností od vrcholu *A*.

Složitost algoritmu

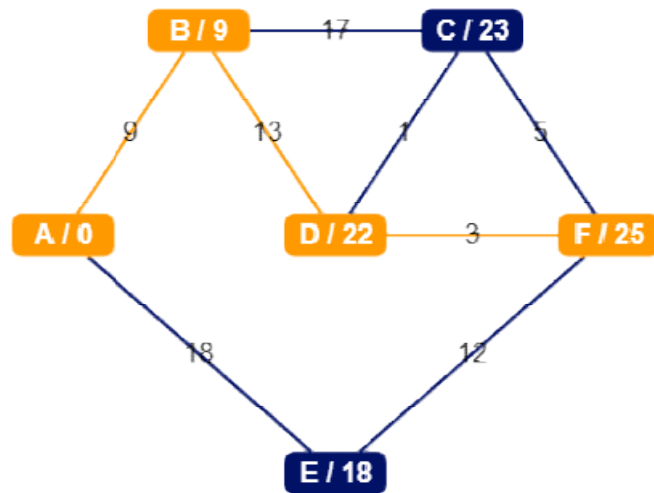
Složitost Dijkstrova algoritmu je $O(|V|^2)$ nebo $O(|E| \cdot \log_2 |V|)$ v závislosti na volbě datové struktury. [29]

Využití algoritmu

Využití Dijkstrova algoritmu je např. v řízení provozu počítačových sítí, v geografických informačních systémech, plánování optimální trasy v logistice apod. [30]

Ukázka algoritmu

Startovací vrchol ukázky je *A*. Obrázek 14 znázorňuje hodnoty po provedení Dijkstrova algoritmu a jednu zvolenou nejkratší cestu mezi vrcholy *A* a *F* (váha 25).



Obrázek 14 Dijkstrův algoritmus (zdroj: vlastní zpracování)

V tabulce 7 je postupný průběh ukázky včetně aktuálních hodnot mezi vrcholem *A* a všemi ostatními vrcholy.

Tabulka 7 Dijkstrův algoritmus
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Prozkoumané vrcholy	Znamé vrcholy	Neznámé vrcholy	A	B	C	D	E	F
1.	A	BE	CDF	0	9	∞	∞	18	∞
2.	AB	CDE	F	0	9	26	22	18	25
3.	ABD	CEF		0	9	23	22	18	25
4.	ABDC	EF		0	9	23	22	18	25
5.	ABDCF	E		0	9	23	22	18	25
6.	ABCDFE			0	9	23	22	18	25

3.2.8 Edmonds-Johnsonův algoritmus

Jak je uvedeno v [11] Edmonds-Johnsonův algoritmus je určen k procházení labyrintu. Labyrint lze reprezentovat grafem, kde křižovatky labyrintu jsou vrcholy a chodby představují hrany.

Následující formulace algoritmu je dle [11]:

Kroky algoritmu

VSTUP: Graf $G(V,E)$ reprezentující labyrint.

ÚKOL: projdi labyrint (každou hranu dvakrát, jednou v každém směru)

KROK 1: Zvol libovolný počáteční vrchol A .

KROK 2: Dokud není každá hrana projitá dvakrát, opakuj procházení podle následujících pravidel:

E1: Každou hranou můžeme projít v jednom směru nejvýše jednou.

E2: Po hraně, po které jsme do vrcholu přišli poprvé, se můžeme vracet pouze tehdy, nemáme-li jinou možnost.

E3: Z hran, které máme k dispozici k odchodu z vrcholu, upřednostňujeme tu hranu, kterou jsme dosud nikdy nešli.

Složitost algoritmu

Časová složitost Edmonds-Johnsonova algoritmu je $O(|V|+|E|)$, každá hrana je procházena právě dvakrát, počet návštěv každého vrcholu je $2 \cdot \text{deg}(V)$. [9]

Využití algoritmu

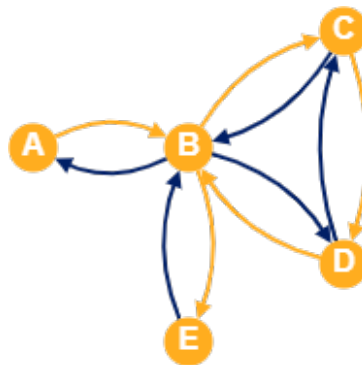
Využití procházení labyrintu lze nalézt všude tam, kde lze projít každou trasu dvakrát. Typickým příkladem jsou dvě strany silnice. Tedy například průjezd kropicího vozu, trasa poštovního doručovatele apod.[10]

Ukázka algoritmu

V příkladu jsou neorientované hrany nahrazeny dvěma orientovanými hranami, jednou v každém směru, viz obrázek 15. Tak je možné lépe znázornit průchod labyrintem.

Do datové struktury LIFO je uložený vrchol při každém průchodu. Vrcholy jsou uloženy v pořadí $A, B, C, D, B, E, B, D, C, B, A$.

Pořadí hran při průchodu labyrintem (obrázek 15), tučně je znázorněn zpětný průchod: $AB, BC, CD, DB, BE, \mathbf{EB, BD, DC, CB, BA}$.



Obrázek 15 Průběh Edmonds-Johnsonova algoritmu po pěti krocích
(zdroj: vlastní zpracování)

3.2.9 Trémauxův algoritmus

Podobně jako Edmonds-Johnsonův algoritmus je i Trémauxův algoritmus určen k procházení labyrintu [31]. Princip algoritmu vychází z prohledávání do hloubky. Do zásobníku je ukládána cesta od počátečního k aktuálnímu vrcholu. Na křižovatkách je směr prohledávání vybírán s dodržением tří pravidel. Od Edmonds-Johnsonova algoritmu se liší jen třetí z nich.[11]

Kroky algoritmu

VSTUP: Graf $G(V,E)$ reprezentující labyrint.

ÚKOL: projít labyrint (každou hranu dvakrát, jednou v každém směru)

KROK 3: Zvol libovolný počáteční vrchol A .

KROK 4: Dokud není každá hrana projitá dvakrát, opakuj procházení podle následujících pravidel:

T1: Každou hranou můžeme projít v jednom směru nejvýše jednou.

T2: Po hraně, po které jsme do vrcholu přišli poprvé, se můžeme vracet pouze tehdy, nemáme-li jinou možnost.

T3: Jestliže přijdeme hranou, procházenou poprvé, do známého vrcholu, tak se hned v následujícím kroku touto hranou vracíme zpět.

Složitost algoritmu

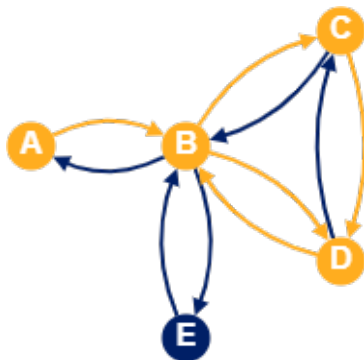
Časová složitost Trémauxova algoritmu je $O(|V|+|E|)$, každá hrana je procházena právě dvakrát, počet návštěv každého vrcholu je $2 \cdot \text{deg}(V)$. [9]

Využití algoritmu

Podobně jako Edmonds-Johnsonův algoritmus i Trémauxův algoritmus se v praxi uplatní v případech, kdy je nutné v nějaké síti procházet jednu hranu dvakrát (tam i zpět).[11]

Ukázka algoritmu

Pro ukázkou průchodu labyrintem byl zvolen graf na obrázku 16. Průchod labyrintem je ale jiný než u Edmonds-Johnsonova algoritmu. Do datové struktury LIFO je každý vrchol uložen právě jednou. Vrcholy jsou uloženy v pořadí: *A, B, C, D, E*. Hrany jsou procházeny v následujícím pořadí (tučně je vyznačen zpětný průchod): *AB, BC, CD, DB, **BD, DC, CB**, BE, **EB**, BA*.



Obrázek 16 Průběh Trémauxova algoritmu po pěti iteracích (zdroj: vlastní zpracování)

Následující algoritmy jsou založeny na prohledávání do hloubky a do šířky.

3.2.10 Zjištění souvislosti grafu

Princip algoritmu pro zjištění souvislosti grafu spočívá v jednom spuštění prohledávání grafu. Pokud prohledávání projde všechny vrcholy grafu, je graf souvislý, v opačném případě ne.[2]

Následující formulace algoritmu je dle[9]:

Kroky algoritmu

VSTUP: graf $G(V, E)$

ÚKOL: zjistit, jestli je graf souvislý

POMOCNÁ PROMĚNNÁ: počet prohledaných vrcholů n .

KROK 1: Zvol *libovolně* počáteční vrchol A .

KROK 2: Nastav $n = 0$.

KROK 4: Spusť prohledávání (do hloubky nebo do šířky) z vrcholu A , v každém vrcholu inkrementuj n .

KROK 5: Zjisti, jestli je n menší než počet vrcholů grafu.

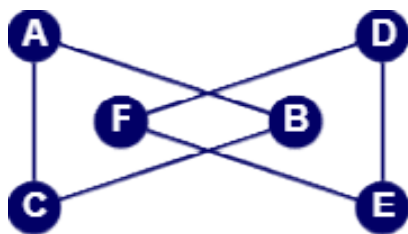
KROK 6: Pokud ano, graf není souvislý, v opačném případě je.

Složitost algoritmu

Časová složitost algoritmu testování souvislosti grafu je $O(|V|+|E|)$. [2]

Ukázka algoritmu

Bylo zvoleno prohledávání do šířky, start prohledávání je ve vrcholu A , viz obrázek 17.



Obrázek 17 Zjišťování souvislosti grafu (zdroj: vlastní zpracování)

Průběh prohledávání demonstruje tabulka 8. Počet prohledaných vrcholů (3) je menší než počet vrcholů grafu (6). Graf tedy není souvislý.

Tabulka 8 Zjišťování souvislosti grafu
(zdroj vlastní zpracování, nástroj MS Excel)

Iterace	Fronta	Navštívené	Nenavštívené
1.	A		ABCDEF
2.	BC	A	BCDEF
3.	C	AB	CDEF
4.		ABC	DEF

3.2.11 Zjištění počtu komponent grafu

Algoritmus pro zjištění počtu komponent ideově vychází z prohledávání grafu (lhostejno jestli do hloubky nebo do šířky). Princip spočívá v opakovaném spuštění prohledávání grafu z dosud neprohledaných vrcholů. Každé ukončené prohledávání představuje jednu komponentu.[2]

Následující formulace algoritmu je dle[9]:

Kroky algoritmu

VSTUP: graf $G(V, E)$

ÚKOL: zjistit, jestli je graf souvislý

POMOCNÁ PROMĚNNÁ: počet prohledaných vrcholů n , počet komponent k , množina prohledaných vrcholů.

KROK 1: Zvol libovolně počáteční vrchol A .

KROK 2: Nastav $n = 0, k = 0$.

KROK 3: Nastav vrchol A jako aktuální vrchol.

KROK 4: Dokud není n rovno počtu vrcholů v grafu, opakuj:

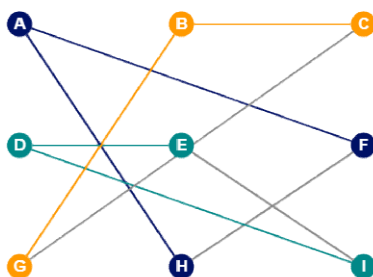
- a. Spusť prohledávání (do hloubky nebo do šířky) z aktuálního vrcholu, v každém vrcholu inkrementuj n , každý prohledaný vrchol vlož do množiny prohledaných vrcholů.
- b. Inkrementuj k .
- c. Nastav jako aktuální vrchol libovolný vrchol, který není v množině prohledaných vrcholů

Složitost algoritmu

Složitost algoritmu zjišťování počtu komponent je $O(|V|+|E|)$. Algoritmus projde všechny vrcholy a hrany právě jednou.[2]

Ukázka algoritmu

Zjišťování počtu komponent bylo ukázáno na grafu z obrázku 18.



Obrázek 18 Zjišťování počtu komponent (zdroj: vlastní zpracování)

Průběh algoritmu a postupné zvyšování počtu komponent je znázorněno v tabulce 9. Bylo použito prohledávání do šířky, tedy fronta LIFO.

Tabulka 9 Zjišťování počtu komponent grafu (zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Fronta	Prozkoumané vrcholy	Neprozkoumané vrcholy	Počet komponent
1.	A		ABCDEFGHI	1
2.	BC	A	BCDEFGHI	
3.	C	BC	CDEFGHI	
4.		ABC	DEFGHI	
5.	D			
6.	EF	ABC D	EFGHI	
7.	F	ABC DE	FGHI	
8.		ABC DEF	GHI	2
9.	G			
10.	HI	ABC DEF G	HI	
11.	I	ABC DEF GH	I	
12.		ABC DEF GHI		3

3.2.12 Vrcholy v komponentě

Algoritmus testuje, zda jsou dva vrcholy v jedné komponentě souvislosti grafu. Algoritmus je založen na prohledávání grafu. Princip spočívá ve spuštění prohledávání z jednoho z testovaných vrcholů. Pokud se v průběhu testování dostaneme do druhého z testovaných vrcholů, leží oba vrcholy v téže komponentě. Pokud ne, v téže komponentě neleží.[2]

Následující formulace algoritmu je dle[9]:

Kroky algoritmu

VSTUP: graf $G(V, E)$, dva testované vrcholy A a B .

ÚKOL: zjistit, vrcholy A a B leží v jedné komponentě

KROK 1: Spust' prohledávání (do hloubky nebo do šířky) vrcholu A .

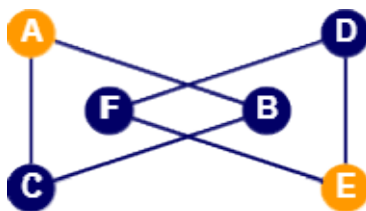
KROK 2: Pokud bude v průběhu prohledávání procházen vrchol B , vrcholy leží v jedné komponentě, v opačném případě neleží.

Složitost algoritmu

Složitost algoritmu testování dvou vrcholů v jedné komponentě je $O(|V|+|E|)$. Jde o prohledávání grafu, počet průchodů je $|V|+|E|$ v dané komponentě. [2]

Ukázka algoritmu

Ukázka algoritmu testování dvou vrcholů A a E v jedné komponentě je zahájena prohledáváním do šířky z vrcholu A (fronta LIFO), viz obrázek 19.



Obrázek 19 Vrcholy v jedné komponentě (zdroj: vlastní zpracování)

Průběh algoritmu je následující, viz také tabulka 10.

- Start prohledávání z vrcholu A .
- Prohledány vrcholy ABC .
- Vrchol E není mezi prohledanými vrcholy.
- Závěr: Vrcholy A a E neleží v jedné komponentě.

Tabulka 10 Vrcholy v jedné komponentě
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Fronta	Navštívené	Nenavštívené
1.	A		ABCDEF
2.	BC	A	BCDEF
3.	C	AB	CDEF
4.		ABC	DEF

3.2.13 Most

Algoritmus určí, zda je testovaná hrana most. Princip spočívá v odebrání testované hrany z grafu a spuštění vyhledávání z jednoho z jejích vrcholů. Pokud se při prohledávání dostaneme do druhého z vrcholů, testovaná hrana most není. V opačném případě most je.[1]

Následující formulace algoritmu je dle[9]:

Kroky algoritmu

VSTUP: graf $G(V, E)$, testovaná hrana AB .

ÚKOL: zjistit, jestli je testovaná hrana most

KROK 1: Odstraň fiktivně hranu AB z grafu G

KROK 2: Spust' prohledávání (do hloubky nebo do šířky) vrcholu A .

KROK 3: Pokud bude v průběhu prohledávání procházen vrchol B , hrana AB není most, v opačném případě je.

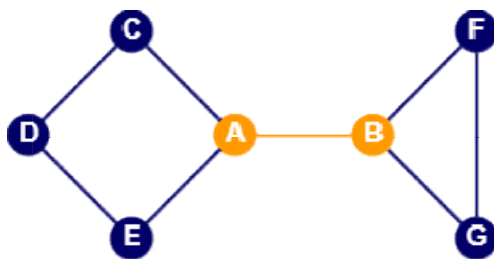
Složitost algoritmu

Složitost algoritmu, který testuje, zda je hrana most, je $O(|V|+|E|)$. Jde opět o prohledávání grafu, počet průchodů je $|V|+|E|$. [2]

Ukázka algoritmu

Na obrázku 20 je graf, který byl zvolen pro ukázkou. Průběh algoritmu je následující:

- Je fiktivně odstraněna testovaná hrana AB z grafu.
- Je spuštěno prohledávání z vrcholu A .
- Prohledání prochází vrcholy C, D a E (neprošlo vrcholem B).
- Závěr: hrana AB je mostem.



Obrázek 20 Testování, zda je hrana most (zdroj: vlastní zpracování)

3.2.14 Délka nejkratší cesty

Algoritmus nalezne délku nejkratší cesty mezi dvěma vrcholy neohodnoceného neorientovaného grafu. Algoritmus spustí prohledávání do šířky z jednoho z testovaných vrcholů. Hloubka, v níž se je prohledán druhý testovaný vrchol, určuje vzdálenost nejkratší cesty. [30]

Následující formulace algoritmu je dle[9]:

Kroky algoritmu

VSTUP: neohodnocený souvislý graf $G(V, E)$, testované vrcholy A a B .

ÚKOL: zjistit délku nejkratší cesty

POMOCNÁ PROMĚNNÁ: seznam prohledaných vrcholů a jejich předchůdců, délka cesty d .

KROK 1: Nastav $d = 0$.

KROK 2: Spust' prohledávání do šířky vrcholu A .

KROK 3: Dokud není dosaženo vrcholu B , opakuj:

- a. po průchodu každé úrovně inkrementuj hodnotu d
- b. pro každý vrchol ulož jeho předchůdce

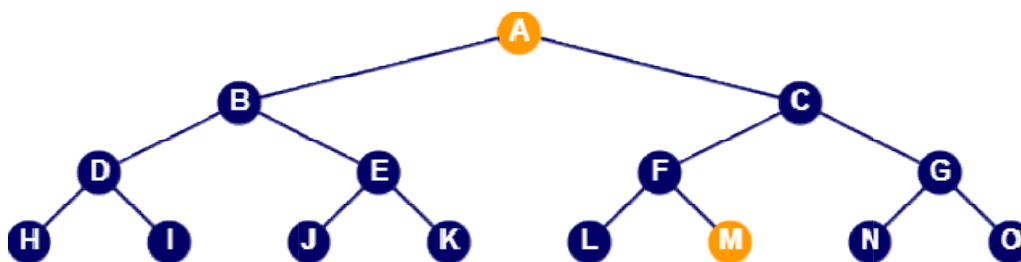
KROK 4: Sledováním předchůdců lze získat konkrétní cestu od vrcholu B do vrcholu A .

Složitost algoritmu

Složitost algoritmu hledání nejkratší cesty je $O(|V|+|E|)$, protože jde o prohledávání do šířky.[2]

Ukázka algoritmu

Na obrázku 21 je graf zvolený pro ukázkou algoritmu hledání nejkratší cesty.



Obrázek 21 Hledání délky nejkratší cesty (zdroj: vlastní zpracování)

V tabulce 11 je znázorněn průběh algoritmu.

- Hledáme nejkratší cestu z vrcholu A do vrcholu M
- Vrchol A je startovací vrchol
- Hledaný vrchol M byl nalezen v hloubce 4, nejkratší vzdálenost vrcholů AM je tedy 3.
- Předchůdcem vrcholu M je vrchol F (prohledán v hloubce 3).
- Předchůdcem vrcholu F je vrchol B (prohledán v hloubce 2).
- Předchůdcem vrcholu B je startovací vrchol A.
- Nejkratší cesta z vrcholu A do vrcholu M je tedy ABFM.

Tabulka 11 Hledání nejkratší cesty mezi dvěma vrcholy
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Vrcholy aktuální úrovně (předchůdce)	Vzdálenost
1.	A(-)	0
2.	B(A), C(A)	1
3.	D(B), E(B), F(C), G(C)	2
4.	H(D), I(D), J(E), K(E), L(F), M(F), N(G), O(G)	3

3.2.15 Bipartita

Algoritmus zjistí, jestli je testovaný graf bipartitní. Princip spočívá v střídavém obarvování vrcholů barvou 1 a barvou 2. Pokud dojde ke kolizi, obsahuje graf kružnici liché délky, a graf není tudíž bipartitní. V opačném případě bipartitní je. [15], [27]

Následující formulace algoritmu je dle [9]:

Kroky algoritmu

VSTUP: neohodnocený souvislý graf $G(V, E)$

ÚKOL: zjistit, jestli je graf bipartitní

KROK 1: Zvol libovolně počáteční vrchol A a obarvi ho barvou B1.

KROK 2: Spust' prohledávání grafu G ze zvoleného vrcholu A.

KROK 3: Dokud nejsou prohledány všechny vrcholy a hrany, opakuj:

- a. Pokud je aktuálně prohledávaný vrchol označen barvou B1, označ všechny jeho sousedy barvou B2, v opačném případě označ všechny jeho sousedy barvou B2.

- b. Pokud je již označovaný vrchol označen jinou barvou, graf není bipartitní, prohledávání ukonči.

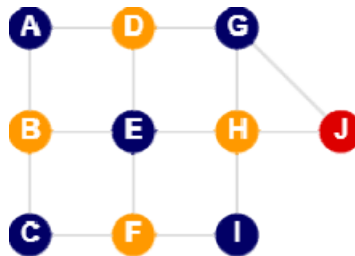
KROK 4: Pokud v průběhu celého prohledávání ke kolizi nedojde, graf je bipartitní.

Složitost algoritmu

Složitost algoritmu testování bipartity je $O(|V|+|E|)$, jde opět o prohledávání grafu.[2]

Ukázka algoritmu

Obrázek 22 znázorňuje testovaný graf pro zjišťování bipartity.



Obrázek 22 Testování bipartity (zdroj: vlastní zpracování)

V tabulce 12 je patrný průběh testování.

- V osmém kroku má být vrchol J obarven barvou B1. Již je ale obarven barvou B2 (krok 6).
- Graf G tedy není bipartitní.

Tabulka 12 Testování bipartity – průběh algoritmu
(zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Aktuální vrchol	Sousední vrcholy	Hloubka	Barva sousedů
1.	A	BD	1	B2
2.	B	EC	2	B1
3.	D	EG	2	B1
4.	C	F	3	B2
5.	E	FH	3	B2
6.	G	HJ	3	B2
7.	F	I	4	B1
8.	H	J	4	B1 (kolize)

3.2.16 Vrchol na kružnici

Algoritmus otestuje, jestli je zadaný vrchol na kružnici. Z inkriminovaného vrcholu je spuštěno prohledávání do šířky. Každý podstrom, jehož kořen je sousedem testovaného vrcholu, je označen jinou barvou. Pokud je v průběhu prohledávání nalezena nestromová hrana s různobarevnými vrcholy, leží testovaný vrchol na kružnici. V opačném případě na kružnici neleží.[32]

Následující formulace algoritmu je dle[9]:

Kroky algoritmu

VSTUP: graf $G(V, E)$, testovaný vrchol A .

ÚKOL: zjistit, jestli vrchol A leží na kružnici

KROK 1: Spust' prohledávání do šířky z testovaného vrcholu

KROK 2: Označ různou barvou všechny vrcholy každého podstromu, jejichž kořeny jsou sousedi vrcholu A .

KROK 3: Pokud v průběhu testování narazíš na nestromovou hranu, otestuj, jestli leží v různých podstromech (její vrcholy jsou označeny různou barvou). Pokud ano, leží vrchol A na kružnici.

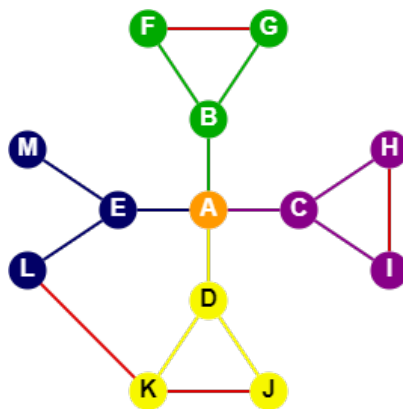
KROK 4: Pokud během celého prohledávání hranu spojující dva podstromy nenajdeš, vrchol A na kružnici neleží.

Složitost algoritmu

Složitost algoritmu testování, zde vrchol leží na kružnici, je $O(|V|+|E|)$, jde o prohledávání do šířky.[2]

Ukázka algoritmu

Algoritmus je ukázán na grafu z obrázku 23.



Obrázek 23 Vrchol na kružnici (zdroj: vlastní zpracování)

Přehled jednotlivých úrovní prohledávání do šířky je v tabulce 13. Startovací vrchol je A . Hrana KL v třetí úrovni spojuje dva různé podstromy, proto vrchol A leží na kružnici.

Tabulka 13 Vrchol na kružnici
(zdroj: vlastní zpracování, nástroj MS Excel)

Úroveň	Prohledané vrcholy	Stromové hrany	Nestromové hrany
1.	A	AB, AC, AD, AE	
2.	BCDE	BF, BG, CH, CI, DJ, DK, EL, EM	
3.	FGHIJKLM		HI, FG, KL

3.2.17 Hrana na kružnici

Algoritmus ověří, jestli je zadaná hrana na kružnici. Je spuštěno prohledávání do šířky z jednoho vrcholu testované hrany. Vrcholy dvou podstromů prohledávání, které mají kořeny vrcholy zkoumané hrany, označuj každý jinou barvou. Pokud je při prohledávání nalezena nestromová hrana s odlišně obarvenými vrcholy, leží testovaná hrana na kružnici. V opačném případě neleží. [2]

Následující formulace algoritmu je dle[9]:

Kroky algoritmu

VSTUP: graf $G(V, E)$, testovaná hrana AB .

ÚKOL: zjistit, jestli hrana AB leží na kružnici

KROK 1: Spust' prohledávání do šířky z vrcholu hrany A

KROK 2: Označ jednou barvou všechny vrcholy podstromu, který obsahuje vrchol B , druhou barvou vrcholy všech ostatních podstromů.

KROK 3: Pokud v průběhu testování objevíš nestrómovou hranu, jejíž vrcholy jsou nestejně označeny, hrana AB leží na kružnici.

KROK 4: Pokud během celého prohledávání takovou hranu nenajdeš, hrana AB na kružnici neleží

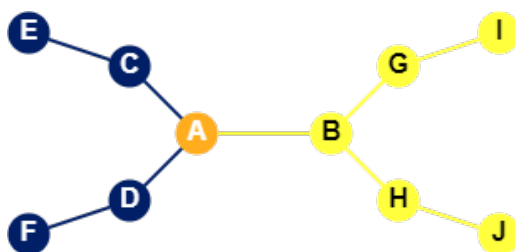
Složitost algoritmu

Složitost algoritmu testování, zda hrana leží na kružnici, je stejně jako u předchozího algoritmu $O(|V|+|E|)$, opět jde o prohledávání do šířky.[2]

Ukázka algoritmu

Na obrázku 24 je graf, jehož hrana AB má být otestována.

- Testovaná hrana AB .
- Zahájeno prohledávání z vrcholu A .
- Všechny vrcholy stromu s kořenem B označuj žlutě, vrcholy podstromu s kořenem A označuj modře.
- V průběhu prohledávání nebyla nalezena nestrómová hrana s různě označenými vrcholy. Proto hrana AB neleží na kružnici.



Obrázek 24 Hrana na kružnici (zdroj: vlastní zpracování)

3.2.18 Nalezení artikulací v grafu

Algoritmus nalezne artikulace v souvislém neorientovaném neohodnoceném grafu. Je založen na prohledávání do hloubky a porovnávání pořadí průchodu jednotlivých vrcholů a jejich následovníků ve vzniklém stromu. [33]

Následující formulace algoritmu je dle[25]:

Kroky algoritmu

VSTUP: souvislý neorientovaný neohodnocený graf $G(V, E)$

ÚKOL: zjistit artikulace v grafu

POMOCNÁ PROMĚNNÁ: množina artikulací K

KROK 1: Spust' prohledávání do hloubky v grafu G z libovolného vrcholu.

KROK 2: Pro každý vrchol V grafu G ulož hodnotu $dfnum(V)$, což je pořadí navštívení vrcholu při prohledávání.

KROK 3: Pro každý vrchol V grafu G spočítej hodnotu $low(V)$ jako minimum z:

- $dfnum(V)$,
- $dfnum(S)$ všech sousedů S vrcholu V přes nestromovou hranu,
- $low(P)$ všech potomků vrcholu V

KROK 4: Do množiny artikulací vlož všechny vrcholy V (kromě kořene stromu), které mají takového potomka P , že platí: $low(P) \geq dfnum(V)$.

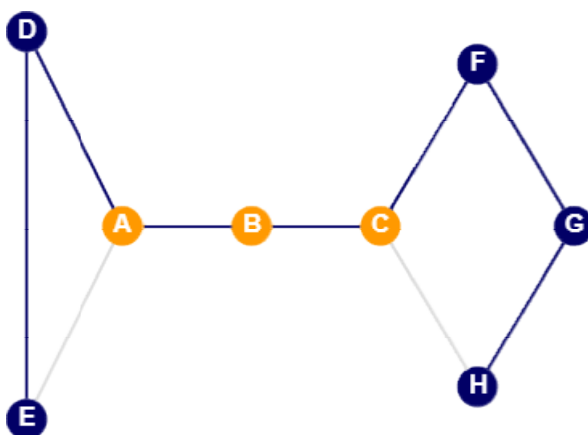
KROK 7: Do množiny artikulací vlož kořen stromu, pokud má alespoň dva potomky.

Složitost algoritmu

Složitost algoritmu nalezení artikulací, je $O(|V|+|E|)$, jde o prohledávání do hloubky. [33]

Ukázka algoritmu

Artikulace jsou hledány v grafu na obrázku 25.



Obrázek 25 Nalezení artikulací v grafu (zdroj vlastní zpracování)

Průběh algoritmu je ukázán v tabulce 14.

- Je spuštěno prohledávání do hloubky z vrcholu A.
- Pro každý vrchol je zaznamenáno pořadí průchodu.

- Dále hledáme pro každý vrchol minimum dle postupu algoritmu.
- Nalezené hodnoty jsou uvedeny v tabulce 14.

Tabulka 14 Nalezení artikulací v grafu (zdroj: vlastní zpracování, nástroj MS Excel)

Iterace	Pořadí prohledávání	dfnum(V)	low(V)	Artikulace
1.	A	1	1	Ano
2.	B	2	2	Ano
3.	C	3	3	Ano
4.	F	4	3	
5.	G	5	3	
6.	H	6	3	
7.	D	7	1	
8.	E	8	1	

3.2.19 Nalezení bloků v grafu

Algoritmus nalezne bloky v souvislém neorientovaném neohodnoceném grafu. Je vlastně rozšířením algoritmu 3.2.18 pro nalezení artikulací. V průběhu hledání artikulací je ukládáno pořadí hran, jimiž algoritmus prochází. Při zpětném procházení hran od potomků jsou vloženy do jednoho bloku všechny hrany, až do nalezení artikulace. [33]

Následující formulace algoritmu je dle [25] a [2]:

Kroky algoritmu

VSTUP: souvislý neorientovaný neohodnocený graf $G(V, E)$

ÚKOL: zjistit bloky v grafu

POMOCNÁ PROMĚNNÁ: množina artikulací K , množina bloků B .

KROK 1: Spust' v grafu algoritmus na vyhledání artikulací K , viz 3.2.18.

KROK 2: Každou prošlou hranu ulož do pole.

KROK 3: Při zpětném chodu (zjišťování minima) od listů vyprázdní b seznam hran představující blok a opakuj pro každou hranu:

- a. pokud procházený vrchol není artikulace, vlož hranu do b ,
- b. v opačném případě ulož blok b do množiny bloků B a vyprázdní b .

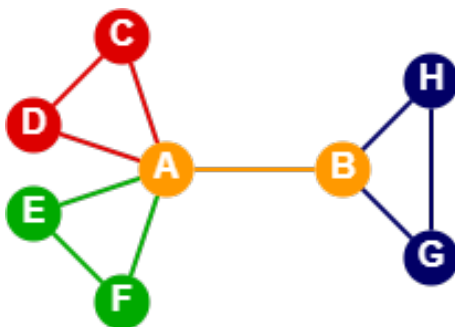
KROK 4: V množině B jsou všechny bloky.

Složitost algoritmu

Složitost algoritmu nalezení bloků, je $O(|V|+|E|)$, stejně jako u hledání artikulací se jedná o prohledávání do hloubky. [33]

Ukázka algoritmu

Pro demonstraci algoritmu hledání bloků v grafu je zvolen graf na obrázku 26.



Obrázek 26 Nalezení bloků v grafu (zdroj: vlastní zpracování)

Průběh algoritmu je znázorněn v tabulce 15. Je spuštěn algoritmus pro hledání artikulací 3.2.18. Při každém nalezení artikulace je definován i blok. Nalezené bloky jsou uvedené také v tabulce 15.

Tabulka 15 Nalezení bloků v grafu (zdroj: vlastní zpracování, nástroj MS Excel)

#	Pořadí průchodu	Hrana	dfnum(V)	low(V) díky ne-stromovým hranám	low(V) díky potomkům	Artikulace
1.	A		1	1	1	Ano
2.	B	AB	2	2	2	Ano
3.	H	BH	3	3	2	
4.	G	HG (+BG)	4	2	2	
Zpět do B, B je artikulace, tj. BGH tvoří blok, dále zpět do A.						
5.	C	AC	5	5	1	
6.	D	CD (+DA)	6	1	1	
Zpět do A, A je artikulace, tj. ACD tvoří blok.						
7.	E		7	8	1	
8.	F		8	1	1	
Zpět do A, A je artikulace, tj. AEF tvoří blok.						
A i B jsou artikulace, proto AB tvoří blok (most)-						

Poznámka: Aplikace bude průběžně doplňována dalšími grafovými algoritmy podle výukových potřeb FIM UHK.

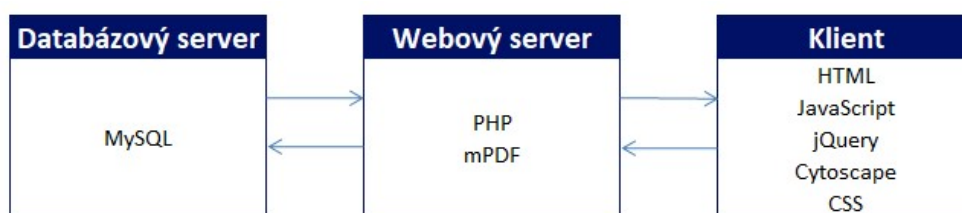
4 Aplikace

4.1 Návrh použitých nástrojů

Návrh aplikace vycházel z požadavků práce a také specifitějších požadavků, které byly na aplikaci kladeny. Těmi jsou zejména všeobecná dostupnost, názornost, přiměřená uživatelská přívětivost a zejména editovatelnost umožňující rozšiřovat obsah a tím i pedagogickou přínosnost aplikace. Volba odpovídající formy aplikace a také vhodných nástrojů pro realizaci byly klíčové pro splnění cílů a výše uvedených požadavků. Proto bylo nutné věnovat této části patřičnou pozornost. Byla proto zvolena webová forma aplikace a dále vybrány odpovídající nástroje umožňující stanovené požadavky splnit. Kromě toho bylo třeba zvolit vhodné nástroje pro další funkce, které práci s aplikací doplňují.[34]

Aplikace pracuje na několika úrovních, viz obrázek 27:

1. na úrovni klienta (webového prohlížeče) – probíhá zde bezprostřední práce s grafem a dalšími daty, animace apod.
2. na úrovni webového serveru – zde je vygenerován výsledný příslušný HTML kód webové stránky
3. na úrovni databázového serveru – zde dochází k manipulaci s daty (ukládání, editace, mazání)



Obrázek 27 Návrh úrovní aplikace (zdroj: vlastní zpracování)

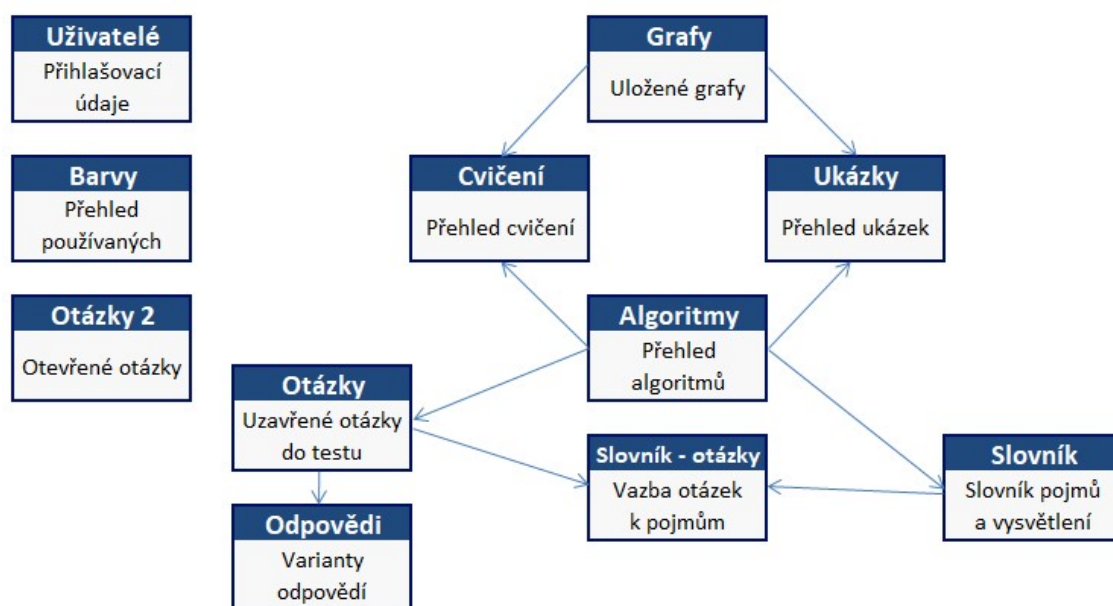
Jako stěžejní nástroj pro vizualizaci grafů a manipulaci s nimi byla zvolena knihovna CytoScape.js. Jde o projekt přidružený k opensourcové platformě Cytoscape, která slouží k vizualizaci molekulárních interakcí. V průběhu dalšího vývoje byly její možnosti rozšířeny obecně pro práci s grafy. Pracuje na bázi JavaScriptu [35]. Dalším zvoleným nástrojem byla knihovna jQuery. Ta pracuje také na bázi JavaScriptu a zjednodušuje manipulaci s daty a událostmi, umožňuje animace apod. [36]

Pro naprogramování kostry celé aplikace, manipulaci s daty, vyřizování požadavků uživatelů a generování kódu na úrovni webového serveru byl zvolen programovací jazyk PHP. [37] Jeho možnosti plně pokrývají potřeby webové aplikace. Jde například o pohodlnou komunikaci s databázovým serverem a všeobecnou dostupnost tematicky zaměřených knihoven. Pro ukládání dat byl zvolen systém řízení báze dat – MySQL. [38] Důvodem volby je zejména spolehlivost a snadná implementovatelnost na různé operační systémy. Layout a design aplikace je koncipován prostřednictvím CSS, výsledný kód je HTML.

Kromě těchto základních nástrojů byly využity dvě PHP knihovny – mPDF umožňuje generování PDF dokumentů a tinyMCE dovoluje formátování ve formulářových polích.

4.2 Návrh databázové struktury

Pro splnění cílů práce a funkcionality aplikace bylo nejdříve nezbytné navrhnout databázovou strukturu a vzájemné vazby tabulek. Na následujícím obrázku 28 je znázorněn E-R diagram. Šipky znázorňují vazbu 1:n.



Obrázek 28 Návrh struktury databáze, E-R diagram (zdroj: vlastní zpracování)

Data v aplikaci jsou uložena v databázi MySQL. Následuje přehled jednotlivých tabulek se stručným popisem obsahu sloupců.

4.2.1 Algoritmy

Struktura databázové tabulky *Algoritmy*, v níž jsou uloženy informace o algoritmech, je uvedena v tabulce 16.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	smallint(5)		UNSIGNED	Ne	Žádná
2.	Nazev	varchar(100)	utf8_czech_ci		Ne	Žádná
4.	Perex	text	utf8_czech_ci		Ano	NULL
5.	Popis	text	utf8_czech_ci		Ano	NULL
6.	hodnoceni	tinyint(1)			Ne	1
7.	hotovo	tinyint(1)			Ne	0

Tabulka 16 Položky tabulky algoritmy (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč
- nazev – název algoritmu
- perex – odstavec se stručným popisem algoritmu
- popis – podrobný popis algoritmu
- hodnoceni – příznak, zda u ukázek zobrazovat hodnocení grafu (aby šlo jeden graf použít pro více algoritmů)
- hotovo – příznak, zda je algoritmus hotova zveřejněn

4.2.2 Grafy

V tabulce *Grafy*, jsou informace o jednotlivých grafech. Jde o umístění uzlů a hran, jejich barvy, popis vrcholů, ohodnocení hran apod., viz tabulka 17.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	int(10)		UNSIGNED	Ne	Žádná
2.	id_u	int(10)		UNSIGNED	Ano	NULL
3.	nazev	varchar(60)	utf8_czech_ci		Ne	Žádná
4.	zavreny	tinyint(1)			Ne	0
5.	ukazkovy	tinyint(1)			Ano	1
6.	popis	text	utf8_czech_ci		Ano	NULL

Tabulka 17 Položky tabulky grafy (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč,
- id_u – cizí klíč, id uživatele, který graf uložil,
- nazev – název grafu,
- zavreny – příznak, zda je graf uzavřený pro úpravy,

- ukazkovy – příznak, zda se graf objevuje mezi ukázkami a cvičeními,
- popis – kód grafu,

4.2.3 Ukázky

V tabulce 18 je znázorněna struktura databázové tabulky *Ukazky* – data o ukázkách demonstrujících algoritmy.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	int(11)		UNSIGNED	Ne	Žádná
2.	id_a	smallint(5)		UNSIGNED	Ano	NULL
3.	id_g	int(10)		UNSIGNED	Ne	Žádná
4.	hotovo	tinyint(1)			Ne	0
5.	informace	text	utf8_czech_ci		Ano	NULL
6.	seznamybody	text	utf8_czech_ci		Ano	NULL
7.	seznamyhran	text	utf8_czech_ci		Ano	NULL

Tabulka 18 Položky tabulky ukázky (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč
- id_a – cizí klíč, id příslušného algoritmu
- id_g – cizí klíč, id příslušného grafu
- hotovo – příznak, zda je ukázka hotová a zveřejněná
- informace – stručná informace pro každý krok
- seznamybody – vypisované seznamy vrcholů pro barvy
- seznamyhran

4.2.4 Cvičení

Databázová struktura tabulky *Cviceni* je uvedena v tabulce 19. Jsou v ní uložena data týkající se příkladů k procvičování.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	int(10)		UNSIGNED	Ne	Žádná
2.	id_a	smallint(5)		UNSIGNED	Ne	Žádná
3.	id_g	int(10)		UNSIGNED	Ne	Žádná
4.	hotovo	tinyint(1)			Ne	0
6.	instrukce	text	utf8_czech_ci		Ano	NULL
7.	posl	text	utf8_czech_ci		Ano	NULL

Tabulka 19 Položky tabulky cvičení (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč
- id_a – cizí klíč, id příslušného algoritmu
- id_g – cizí klíč, id příslušného grafu
- hotovo – příznak, zda je cvičení hotové a zveřejněné
- instrukce – počáteční instrukce pro cvičení
- posl – posloupnost správných kroků

4.2.5 Barvy

Tabulka *Barvy* je určená k uložení kódů barev používaných v ukázkách a cvičeních. Její struktura je popsána v tabulce 20.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	tinyint(11)		UNSIGNED	Ne	Žádná
2.	nazev	varchar(30)	utf8_czech_ci		Ne	Žádná
3.	pismo	varchar(6)	utf8_czech_ci		Ne	fff
4.	kod	varchar(6)	utf8_czech_ci		Ne	Žádná

Tabulka 20 Položky tabulky barvy (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč
- nazev – název barvy
- pismo – hexadecimální kód písma
- kod – hexadecimální kód barvy

4.2.6 Slovník

Ze struktury databázové tabulky *Slovník* je patrné, že slouží k uložení pojmů a jejich objasnění, viz tabulka 21.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	int(10)		UNSIGNED	Ne	Žádná
2.	pojmem	varchar(100)	utf8_czech_ci		Ne	Žádná
3.	vysvetleni	text	utf8_czech_ci		Ne	Žádná
4.	popis_obrazku	varchar(100)	utf8_czech_ci		Ano	NULL
5.	hotovo	tinyint(1)			Ne	0

Tabulka 21 Položky tabulky slovník (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč

- pojem – pojem, který je vysvětlen
- vysvetleni – textové formátované vysvětlení pojmu
- popis_obrazku – text k obrázku, pokud je přiložen
- hotovo – příznak, zda je pojem dokončen a zveřejněn

4.2.7 Otázky

Uložení otázek do testu, které umožňuje uživateli procvičování je uloženo v tabulce *Otázky*. Její struktura je znázorněna v tabulce 22.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	int(10)		UNSIGNED	Ne	Žádná
2.	id_a	smallint(5)		UNSIGNED	Ano	NULL
3.	spravna	int(10)		UNSIGNED	Ano	NULL
4.	zneni	text	utf8_czech_ci		Ano	NULL
5.	hotovo	tinyint(1)			Ne	0

Tabulka 22 Položky tabulky otázky (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč
- id_a – cizí klíč, id algoritmu, pokud se k němu otázka vztahuje
- spravna – id správné odpovědi
- zneni – znění otázky
- hotovo – příznak, zda je otázka dokončená a zveřejněná

4.2.8 Odpovědi

V databázové tabulce *Odpovedi* jsou uloženy odpovědi k testovým otázkám, viz tabulka 23.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	int(11)		UNSIGNED	Ne	Žádná
2.	id_o	int(10)		UNSIGNED	Ne	Žádná
3.	zneni	text	utf8_czech_ci		Ne	Žádná

Tabulka 23 Položky tabulky odpovědi (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč
- id_o – cizí klíč, id otázky, k níž se odpověď vztahuje
- zneni – znění odpovědi

4.2.9 Users

K uložení uživatelů a přihlašovacích údajů byla vytvořena databázová tabulka *Users*. Její strukturu znázorňuje tabulka 24.

	Název sloupce	Typ	Kódování	Vlastnosti	Nulový	Implicitní hodnota
1.	id (primární klíč)	int(10)		UNSIGNED	Ne	Žádná
2.	mail	varchar(100)	utf8_czech_ci		Ne	Žádná
3.	heslo	varchar(32)	utf8_czech_ci		Ne	Žádná
4.	jmeno	varchar(40)	utf8_czech_ci		Ne	Žádná
5.	prijmeni	varchar(60)	utf8_czech_ci		Ne	Žádná

Tabulka 24 Položky tabulky users (zdroj: vlastní zpracování, nástroj MS Excel)

Význam jednotlivých sloupců

- id – primární klíč
- e-mail – přihlašovací e-mail
- heslo – přihlašovací heslo
- jmeno – jméno živatele
- prijmeni – příjmení uživatele

4.3 Implementace

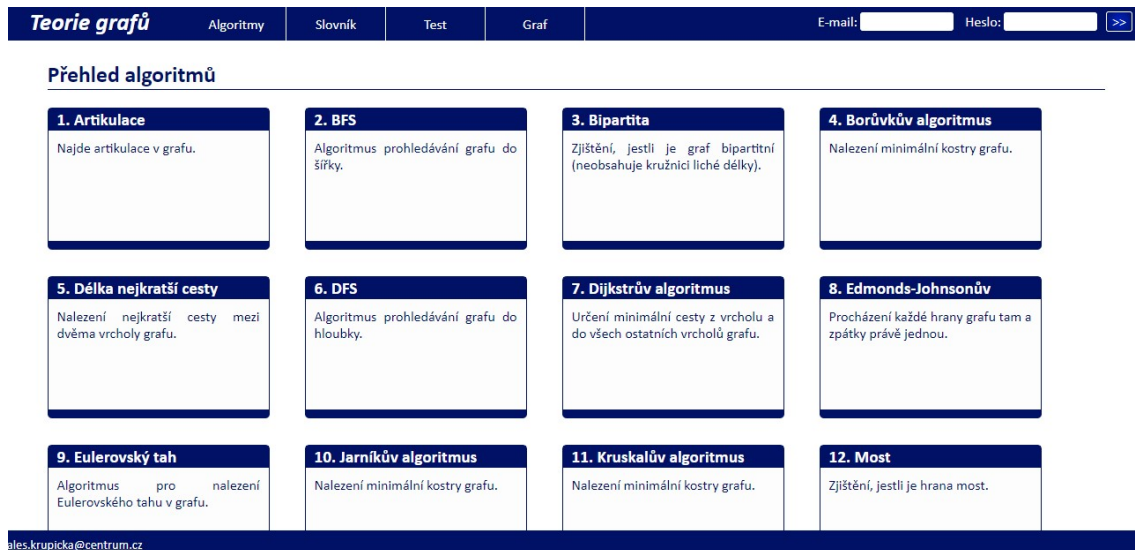
Základní kostra aplikace byla vytvořena v programovacím jazyku PHP. Jde zejména o následující funkce:

- přihlášení uživatele a udržování stavu přihlášení,
- komunikace s databázovým serverem s manipulací s daty,
- reakce na požadavky uživatele a generování HTML kódu,
- využívání knihoven.

Implementace webové aplikace vychází kromě cílů ze dvou základních rolí uživatelů – návštěvníka a administrátora. Administrátor (vyučující) má oprávnění upravovat obsah databáze, konkrétně vkládá a upravuje algoritmy, grafy, přiřazuje jim ukázky a cvičení. Dále ukládá, edituje a maže termíny ve slovníku, otázky a varianty odpovědí, barvy a také ostatní uživatele. Návštěvník (student) využívá vložený obsah ke studiu – může studovat textový obsah, sledovat videa, prohlíží si interaktivní ukázky, cvičení využívá k ověřování pochopení algoritmů. Své znalosti prověřuje v testu. Kromě toho může samostatně vkládat grafy a zkusit jednotlivé algoritmy na nich. Vložený graf může uložit do souboru a následně ho načíst.

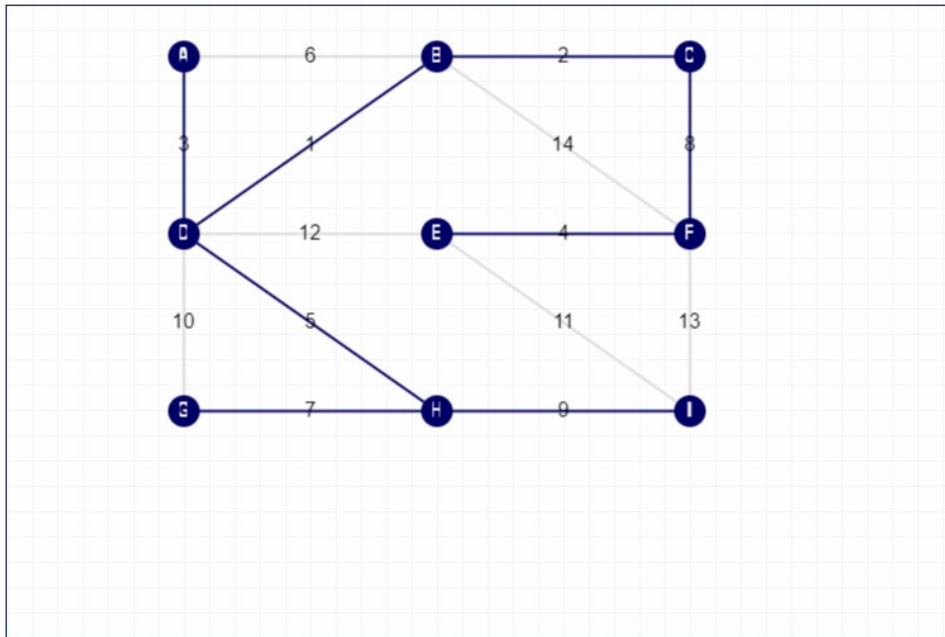
4.4 Grafické rozhraní

Vzhledem k tomu, že jde o webovou aplikaci, je základní rámec grafického rozhraní definován prohlížečem příslušného zařízení. Layout aplikace byl implementován prostřednictvím CSS (Cascade Style Sheet). Na obrázku 29 je titulní obrazovka aplikace s přehledem algoritmů.



Obrázek 29 Úvodní stránka aplikace
(zdroj: vlastní zpracování)

Zásadní funkce – tedy práce s grafem probíhá v definovaném prostoru, v němž jsou prostřednictvím knihovny CytoScape.js a JavaScriptu řešeny reakce na jednotlivé akce uživatele, viz obrázek 30. Na obrázku 31 jsou ovládací prvky, které se nacházejí na stejné stránce.



Obrázek 30 Prostor pro zobrazení grafu a práci s ním
(zdroj: vlastní zpracování)

Editace grafu + 📁 📄 🗑️ 🚩 ☰ 👁️

1.00 750x500

🔗 📄 📄 ?

Žádná akce
Přidat vrcholy
Odebrat vrcholy
Přidat hrany
Odebrat hrany

0 1 2 3 4 5 6 7 8 9

Vlastnosti grafu

- » orientovaný
- » ohodnocený
- » popis vrcholů

Vlastnosti

- » Vrcholů: 9
- » Hran: 14
- » Souvislost: ano
- » Komponent: 1

Ohodnocení hran

AB	6	BC	2	CF	8	FI	13	IH	9	HG	7
GD	10	DA	3	DE	12	EF	4	DB	1	BF	14
DH	5	EI	11								

Popis vrcholů

A B C D E F G H I


Obrázek 31 Ovládací prvky pro graf (zdroj: vlastní zpracování)

4.5 Popis a ovládání aplikace

V této kapitole následuje stručný popis a ovládání aplikace. Podrobné uživatelské příručky pro ovládání aplikace určené pro administrátory a pro studenty jsou k dispozici v příloze a také v elektronické podobě na adrese www.teoriegrafu.cz.

4.6 Ovládání administrátorské části

Využívání administrátorské části aplikace je podmíněno přihlášením do systému prostřednictvím formuláře v pravém horním rohu obrazovky, viz obrázek 32.



The image shows a login form with a dark blue header. It contains two input fields: 'E-mail:' followed by a white text box, and 'Heslo:' followed by a white password box. To the right of the password box is a blue button with a white double arrow symbol '>>'.

Obrázek 32 Přihlašovací formulář (zdroj: vlastní zpracování)

Poznámka: Testovací přihlašovací údaje jsou:

- E-mail: `admin@uhk.cz`
- Heslo: `uhk123`

Po obhájení diplomové práce bude tento obecný administrátor z bezpečnostních důvodů zrušen.

Po úspěšném přihlášení je administrátor přesměrován na rozcestník, odkazující na jednotlivé sekce, viz obrázek 33.



Obrázek 33 Rozcestník administrace (zdroj: vlastní zpracování)

4.6.1 Algoritmy

V sekci algoritmů má administrátor možnost editovat algoritmy, jejich popis a další zobrazované vlastnosti. Jako první je zobrazen název. V následujících sloupcích jsou číselně a symbolicky uvedeny vlastnosti a odkazy na jejich editaci, obrázek 34.

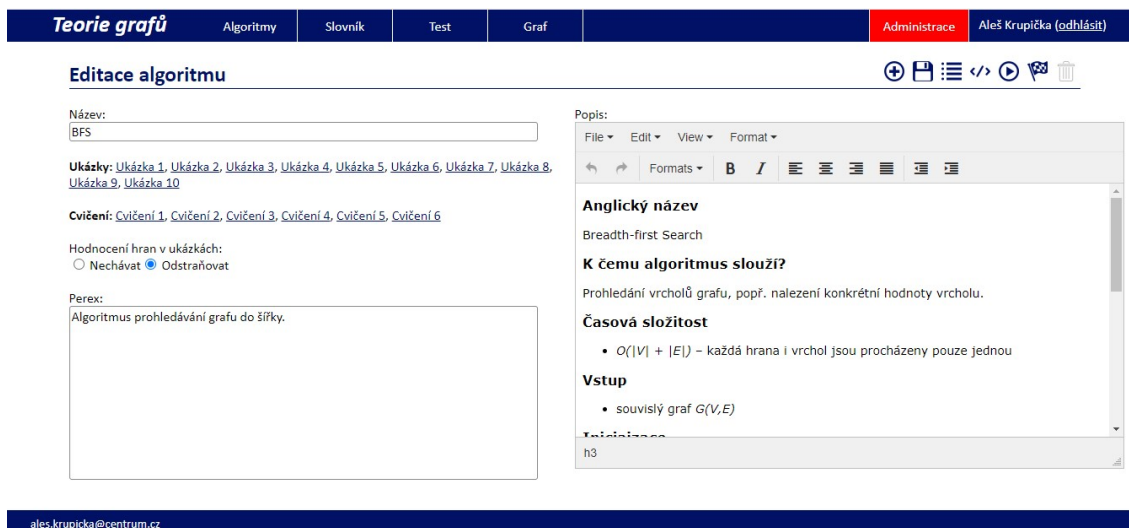
#	Název	Uk.	Cv.	Vid.	Kód	Ex.	Veř.
1.	Artikulace	0	0	⊙	↔	✓	🗑️
2.	BFS	10	6	⊙	↔	✓	🗑️
3.	Bipartita	4	2	⊙	↔	✓	🗑️
4.	Bořůvkův algoritmus	10	8	⊙	↔	✓	🗑️
5.	Délka nejkratší cesty	4	3	⊙	↔	✓	🗑️
6.	DFS	7	6	⊙	↔	✓	🗑️
7.	Dijkstrův algoritmus	2	0	⊙	↔	✓	🗑️
8.	Edmonds-Johnsonův	3	2	⊙	↔	✓	🗑️
9.	Eulerovský tah	4	3	⊙	↔	⊙	🗑️
10.	Jarníkův algoritmus	9	7	⊙	↔	✓	🗑️
11.	Kruskalův algoritmus	9	8	⊙	↔	✓	🗑️
12.	Most	3	2	⊙	↔	✓	🗑️
13.	Počet komponent	3	2	⊙	↔	✓	🗑️
14.	Souvislost grafu	3	2	⊙	↔	✓	🗑️
15.	Tarryho algoritmus	0	0	⊙	↔	✓	🗑️
16.	Trémauxův algoritmus	3	2	⊙	↔	⊙	🗑️
17.	Vrchol na kružnici	3	2	⊙	↔	✓	🗑️
18.	Vrcholy v komponentě	5	4	⊙	↔	✓	🗑️

Obrázek 34 Výpis algoritmů (zdroj: vlastní zpracování)

U algoritmu lze editovat následující položky:

- Název – název algoritmu, řetězec v maximální délce 100 znaků (povinná položka),
- Perex – stručný popis algoritmus, text v délce několika vět.
- Popis – obsáhlejší popis algoritmu, formátovaný delší text, který je vkládán prostřednictvím nástroje TinyMCE.
- Hodnocení hran v ukázkách – příznak, který určuje, zda má v ukázkách zůstat u grafů viditelné hodnocení (ocenění) hran. Zaškrťovací hodnota typu boolean.

Kromě těchto položek je zde výpis ukázek a cvičení vázaných k algoritmu. Přímou u nich je odkaz na jejich případnou editaci, viz obrázek 35.



Obrázek 35 Editační formulář algoritmu
(zdroj: vlastní zpracování)

4.6.2 Grafy

V aplikaci je možné zobrazit přehled grafů, k nimž má uživatel přístup. Jsou rozděleny na dvě skupiny. V první skupině jsou grafy, k nimž se vážou nějaké ukázky nebo cvičení. Ty jsou také přístupné v příslušných sekcích u algoritmů. Ve druhé skupině jsou grafy, které má uživatel pro svoji potřebu, např. k tvorbě prezentací apod. Grafy lze mezi skupinami libovolně přemísťovat kliknutím na symbol oka. Dalším příznakem je nastavení, zda je graf dokončený, a tedy zveřejněný (symbol vlajčky), viz obrázek 36.

# Ukázkové grafy					# Ostatní grafy						
#	Ukázka	Uk.	Cv.	Veř.	Uk.	#	Ukázka	Uk.	Cv.	Veř.	Uk.
1.	Ukázka 02 – kostra	16	16	🚩	👁	18.	Artiklace	0	0	🚩	👁
2.	Ukázka 03 – kostra, prohledávání	25	25	🚩	👁	19.	Artiklace 2	0	0	🚩	👁
3.	Ukázka 06 – prohledávání	4	4	🚩	👁	20.	Artiklace txt	0	0	🚩	👁
4.	Ukázka 08 – prohledávání	9	9	🚩	👁	21.	Bipartita	0	0	🚩	👁
5.	Ukázka 09 – most	1	1	🚩	👁	22.	Bipartita 2	0	0	🚩	👁
6.	Ukázka 10 – most	1	1	🚩	👁	23.	Bipartita txt	0	0	🚩	👁
7.	Ukázka 11 – komponenty	16	16	🚩	👁	24.	Bipartita ukázka	1	0	🚩	👁
8.	Ukázka 12 – komponenty	1	1	🚩	👁	25.	Bloky txt	0	0	🚩	👁
9.	Ukázka 13 – komponenty	4	4	🚩	👁	26.	Borůvka	0	0	🚩	👁
10.	Ukázka 13 – souvislost, prohledávání	1	1	🚩	👁	27.	BSF do textu	2	0	🚩	👁
11.	Ukázka 14 – kostra	16	16	🚩	👁	28.	Cvičení 10	0	0	🚩	👁
12.	Ukázka 15 – kostra, prohledávání	25	25	🚩	👁	29.	Cvičení 11	0	0	🚩	👁
13.	Ukázka 19 – eulerovský tah	1	1	🚩	👁	30.	Cvičení 12	0	0	🚩	👁
14.	Ukázka 21 – eulerovský tah	1	1	🚩	👁	31.	Cvičení 12	0	0	🚩	👁
15.	Ukázka 22 – labyrint	4	4	🚩	👁	32.	Cvičení 5	0	0	🚩	👁
16.	Ukázka 23 – labyrint	4	4	🚩	👁	33.	Cvičení 6	0	0	🚩	👁
17.	Ukázka 24 – kostra	3	0	🚩	👁	34.	Cvičení 9	0	0	🚩	👁
						35.	Délka nejkratší cesty txt	1	0	🚩	👁

Obrázek 36 Přehled grafů (zdroj: vlastní zpracování)

Graf, který není označený jako dokončený, lze editovat kliknutím na příslušný řádek. Editační stránka grafu je rozdělena na dvě části. V levé je samotný graf, vpravo ovládací prvky. Pro lepší umístění jednotlivých vrcholů je graf podložen mřížkou. V menu lze vybrat varianty pro editaci, viz obrázek 37.

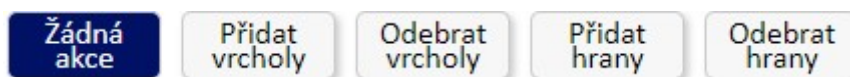


Obrázek 37 Editace grafu (zdroj: vlastní zpracování)

V této části lze editovat základní vlastnosti grafu.

- přidávat a odebírat vrcholy a hrany,
- přemísťovat vrcholy,
- obarvovat vrcholy a hrany,
- přepínat orientaci grafu,
- přepínat a nastavovat ohodnocení hran,
- přepínat popis vrcholů.

Veškerou manipulaci s grafem lze provádět přepnutím do příslušného módu, viz obrázek 38.



Obrázek 38 Nastavení manipulace s vrcholy a hranami (zdroj: vlastní zpracování)

Všechny nastavené vlastnosti je nutné uložit kliknutím na ikonu diskety v pravé horní části okna.

4.6.3 Ukázky

V sekci ukázek může administrátor editovat komentované ukázky algoritmů. Editovanou ukázkou je možné vybírat z přehledu všech ukázek. Ty jsou řazeny po jednotlivých grafech, k nimž jsou ukázky vztaženy. U každé ukázky je příznak vlajky, který značí, zda je ukázka považována za hotovou (a je tedy i zveřejněna) nebo rozpracovanou (šedá vlaječka). Ukázkou lze editovat jen po předchozím přepnutí do rozpracovaného stavu. Rozpracované ukázky jsou řazeny jako první. Rozpracovanou ukázkou lze také smazat pomocí ikonky koše, viz obrázek 39.

Teorie grafů Algoritmy Slovník Test Graf Administrace Aleš Krupička (odhlásit)

Ukázky (42)

Ukázky 02 – kostra	Ukázky 03 – kostra, prohledávání	Ukázky 06 – prohledávání	Ukázky 08 – prohledávání
1. Bipartita	1. Borůvkův algoritmus	1. BFS	1. Délka nejkratší cesty
2. Kruskalův algoritmus	2. DFS	2. DFS	2. DFS
3. Jarníkův algoritmus	3. BFS		3. BFS
4. Borůvkův algoritmus	4. Kruskalův algoritmus		
	5. Jarníkův algoritmus		

Ukázky 09 – most	Ukázky 10 – most	Ukázky 11 – komponenty	Ukázky 12 – komponenty
1. Most	1. Most	1. Počet komponent	1. Počet komponent
		2. Vrcholy v komponentě	
		3. Souvislost grafu	
		4. Vrcholy v komponentě	

Ukázky 13 – komponenty	Ukázky 13 – souvislost, prohledávání	Ukázky 14 – kostra	Ukázky 15 – kostra, prohledávání
1. Vrcholy v komponentě	1. Souvislost grafu	1. Délka nejkratší cesty	1. BFS
2. Vrcholy v komponentě		2. Kruskalův algoritmus	2. DFS
		3. Jarníkův algoritmus	3. Kruskalův algoritmus
		4. Borůvkův algoritmus	4. Jarníkův algoritmus
			5. Borůvkův algoritmus

ales.krupicka@centrum.cz

Obrázek 39 Přehled ukázek (zdroj: vlastní zpracování)

Jak již bylo zmíněno, editovat (a mazat) lze jen ukázkou, která je v rozpracovaném stavu a má zašedlou ikonku dokončení. Editační stránka je rozdělena na dvě části. Vlevo je viditelný samotný graf a vpravo jsou ovládací prvky, viz obrázek 40.

Teorie grafů Algoritmy Slovník Test Graf Administrace Aleš Krupička (odhlásit)

Bipartita (Ukázky 02 – kostra) – krok 1

Informace:
Bipartitu budeme zkoumat prohledáním a střídavým barevným označováním vrcholů.

Barvy:
0 1 2 3 4 5 6 7 8 9

Kroky:
1 2 3 4 5 6

Obrázek 40 Ovládání ukázky (zdroj: vlastní zpracování)

Pomocí ovládacích prvků lze editovat komentáře k jednotlivým krokům a barvy uzlů a hran. Nic z uvedeného není třeba ukládat. Barevné nastavení grafu i komentáře se automaticky ukládají do databáze.

4.6.4 Cvičení

V sekci cvičení může administrátor editovat cvičení sloužící k ověřování znalostí. Editované cvičení je možné vybírat z přehledu všech cvičení. Ta jsou řazena po jednotlivých grafech, k nimž jsou cvičení vztažena. Pokud chce administrátor cvičení editovat, musí je nejdříve přepnout do rozpracovaného stavu. Rozpracovaná cvičení jsou řazena jako první. Rozpracovaná cvičení lze také smazat, viz obrázek 41.

The screenshot shows the 'Teorie grafů' administration interface. At the top, there is a navigation bar with tabs: 'Teorie grafů', 'Algoritmy', 'Slovník', 'Test', 'Graf', 'Administrace', and 'Aleš Krupička (odhlásit)'. Below the navigation bar, the section 'Cvičení (39)' is displayed. The exercises are arranged in a grid of 12 cards, organized into three rows and four columns. Each card has a title, a list of topics, and a status icon (a small square with a checkmark and a trash can icon).

Ukázky 02 – kostra	Ukázky 03 – kostra, prohledávání	Ukázky 06 – prohledávání	Ukázky 08 – prohledávání
1. Bipartita	1. Borůvkův algoritmus	1. BFS	1. Délka nejkratší cesty
2. Kruskalův algoritmus	2. DFS	2. DFS	2. DFS
3. Jarníkův algoritmus	3. BFS		3. BFS
4. Borůvkův algoritmus	4. Kruskalův algoritmus		
	5. Jarníkův algoritmus		
Ukázky 09 – most	Ukázky 10 – most	Ukázky 11 – komponenty	Ukázky 12 – komponenty
1. Most	1. Most	1. Počet komponent	1. Počet komponent
		2. Vrcholy v komponentě	
		3. Souvislost grafu	
		4. Vrcholy v komponentě	
Ukázky 13 – komponenty	Ukázky 13 – souvislost, prohledávání	Ukázky 14 – kostra	Ukázky 15 – kostra, prohledávání
1. Vrcholy v komponentě	1. Souvislost grafu	1. Délka nejkratší cesty	1. BFS
2. Vrcholy v komponentě		2. Kruskalův algoritmus	2. DFS
		3. Jarníkův algoritmus	3. Kruskalův algoritmus
		4. Borůvkův algoritmus	4. Jarníkův algoritmus
			5. Borůvkův algoritmus

ales.krupicka@centrum.cz

Obrázek 41 Výpis cvičení (zdroj: vlastní zpracování)

Jak již bylo zmíněno, editovat (a mazat) lze jen cvičení, která je v rozpracovaném stavu a má zašedlou ikonku dokončení. Editační stránka je rozdělena na dvě části. Vlevo je viditelný samotný graf a vpravo jsou ovládací prvky. Pomocí ovládacích prvků lze editovat instrukce ke cvičení a barvy uzlů a hran, viz obrázek 42.

Teorie grafů Algoritmy Slovník Test Graf Administrace Aleš Krupička (odhlásit)

Borůvkův algoritmus (Ukázky 03 – kostra, prohledávání) – krok 1

Instrukce:
Nalezněte minimální kosteru použitím Borůvkova algoritmu. Začněte ve vrcholu A, klikajte postupně na přidávané vrcholy a hrany. Dodržujte lexikografické pravidlo.

Barvy:
0 1 2 3 4 5 6 7 8 9

Aktuální krok: 1/13
Vrchol A

ales.krupicka@centrum.cz

Obrázek 42 Úprava cvičení (zdroj: vlastní zpracování)

4.6.5 Test

V sekci *Test* je možné editovat testové otázky a odpovědi na ně. Požadovanou testovou otázku je možné vybírat z přehledu všech otázek, viz obrázek 43.

Teorie grafů Algoritmy Slovník Test Graf Administrace Aleš Krupička (odhlásit)

Otázky (67)

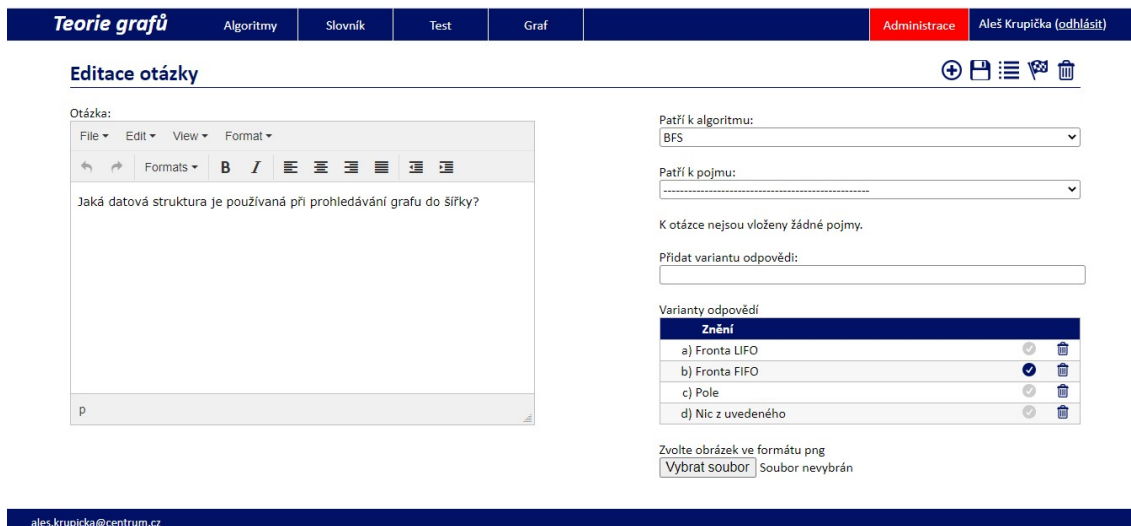
- Jaká datová struktura je používána při prohledávání grafu do šířky?
- Který vrchol bude při prohledávání do šířky prohledán jako třetí, začne-li prohledávání ve vrcholu A?
- Kolik hran tvoří strom prohledávání, který vznikne při prohledávání do šířky grafu, který má 10 vrcholů?
- Kolik hran tvoří strom prohledávání, který vznikne při prohledávání do hloubky grafu, který má 8 vrcholů?
- Kolik hran má úplný graf s šesti vrcholy?
- Kolik hran má úplný bipartitní graf $K_{3,4}$?
- Bipartitní graf je takový graf, jehož množinu vrcholů lze rozdělit na dvě části, přičemž...
- Indukovaný podgraf se vyznačuje tím, že...
- Graf na obrázku je...
- Graf na obrázku je...
- Indukovaný podgraf úplného grafu je...
- Grafy ABCD a EFHG jsou...
- Posloupnost vrcholů a hran, kde vrcholy jsou navzájem různé, se nazývá...
- Jestliže pro každé dva vrcholy x a y existuje cesta z x do y , pak graf označíme jako...
- Které tvrzení o grafu na obrázku není pravdivé?
- Jaká je cena minimální kostry grafu na obrázku?
- Jaká je cena minimální kostry grafu na obrázku?

ales.krupicka@centrum.cz

Obrázek 43 Výpis testových otázek (zdroj: vlastní zpracování)

Měnit lze následující položky, viz obrázek 44:

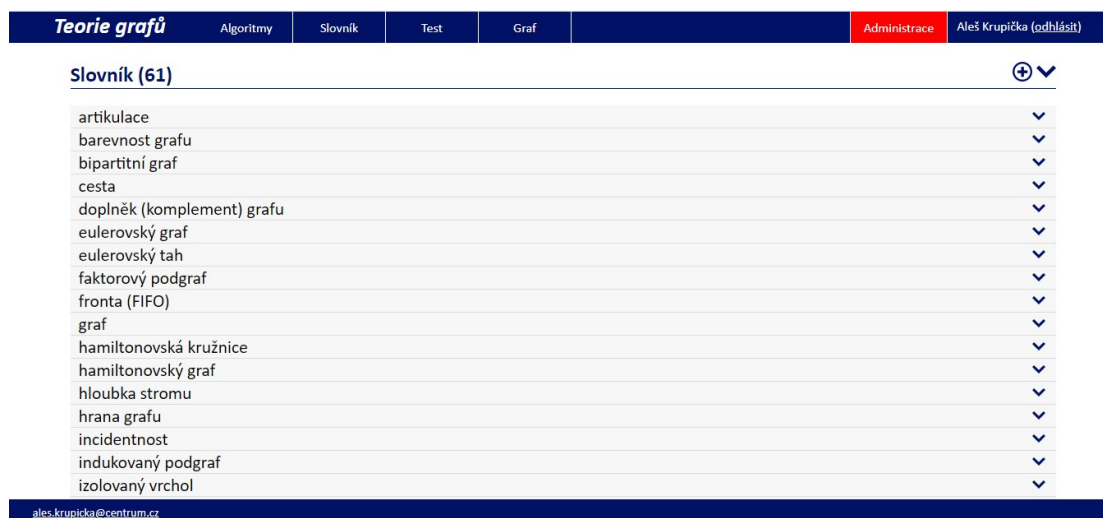
- Znění otázky (včetně formátování pomocí nástroje tinyMCE), jde o textovou položku.
- Přiřazení k algoritmu, výběr ze seznamu – tato nepovinná hodnota zajistí vložení do testování konkrétního algoritmu.
- Obrázek – lze vkládat obrázek typu png, který je přiřazen k otázce.
- Varianty odpovědí – je možné vkládat, editovat a mazat varianty odpovědí



Obrázek 44 Editace testové otázky (zdroj: vlastní zpracování)

4.6.6 Slovník

V sekci *Slovník* může administrátor editovat položky ve slovníku a jejich vysvětlení. Editovaný pojem je možné vybírat z přehledu všech pojmů, viz obrázek 45.



Obrázek 45 Výpis pojmů ve slovníku (zdroj: vlastní zpracování)

Každý pojem je možné kliknutím rozbalit. Tím se zobrazí vysvětlení pojmu. Editovat lze, viz obrázek 46:

- Vysvětlení pojmu (včetně formátování pomocí nástroje tinyMCE), jde o textovou položku.
- Pojem – řetězec o maximální délce 100 znaků.
- Obrázek – lze vkládat obrázek typu png, který je přiřazen k pojmu.
- Popis obrázku – řetězec o maximální délce 100 znaků.

Editace pojmu

Vysvětlení:

File Edit View Format

Formats **B** *I* [text alignment icons]

Cesta v grafu je posloupnost vrcholů a hran $(v_0, e_1, v_1, \dots, e_n, v_n)$, kde vrcholy v_0, \dots, v_n jsou navzájem různé vrcholy grafu G a pro každé $i = 1, 2, \dots, n$ je $e_i = \{v_{i-1}, v_i\}$ je prvkem $E(G)$.

Pojem:

Popis obrázku:

Obrázek:

Obrázek 46 Editace položky ve slovníku (zdroj: vlastní zpracování)

4.6.7 Barvy

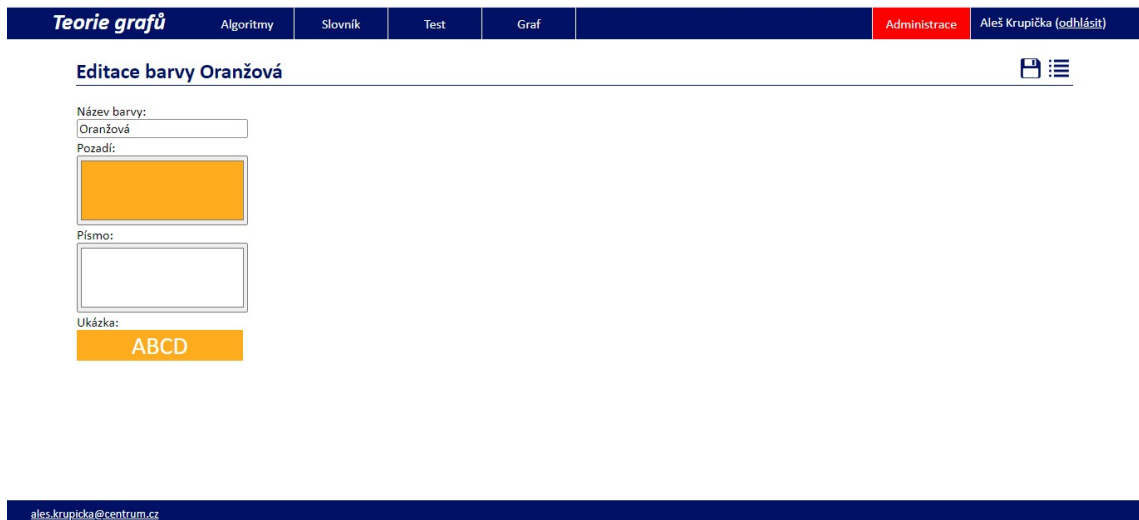
V administrátorské sekci lze také editovat jednotlivé *Barvy*, které pak ovlivní veškerou vizuální podobu grafů. Editaci barev je proto nutné provádět obezřetně. Po otevření sekce barev vidíme jejich přehled včetně vizuální podoby, viz obrázek 47.

Barvy

1. Modrá
2. Zelená
3. Červená
4. Oranžová
5. Fialová
6. Žlutá
7. Modrozelená
8. Hnědá
9. Černá

Obrázek 47 Výpis barev (zdroj: vlastní zpracování)

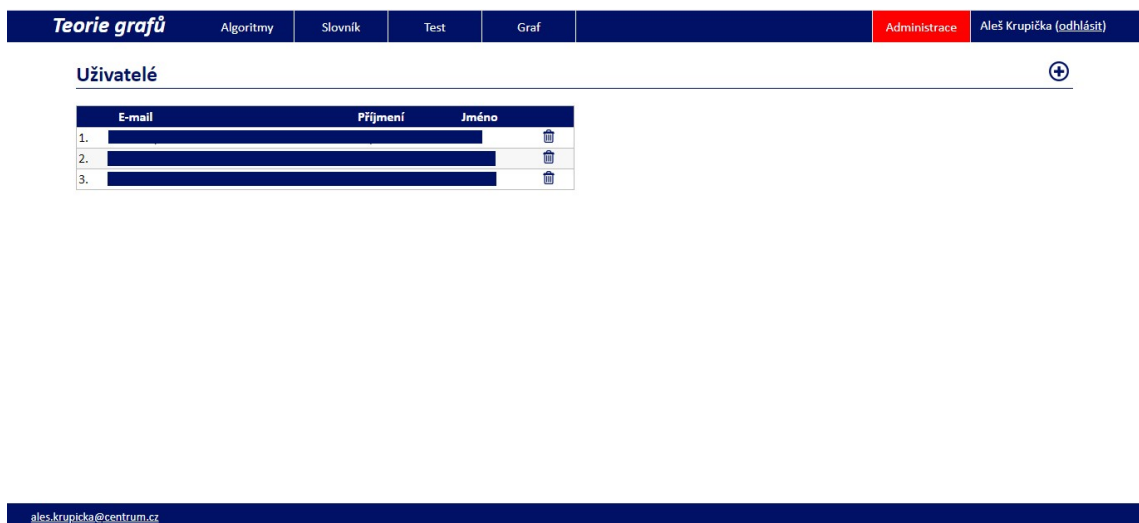
Po výběru barvy je možné ve formuláři upravovat název barvy, barvu pozadí a barvu textu. Barev je celkem deset (z toho devět nastavitelných) a nelze je přidávat ani mazat, viz obrázek 48.



Obrázek 48 Editace Barev (zdroj: vlastní zpracování)

4.6.8 Uživatelé

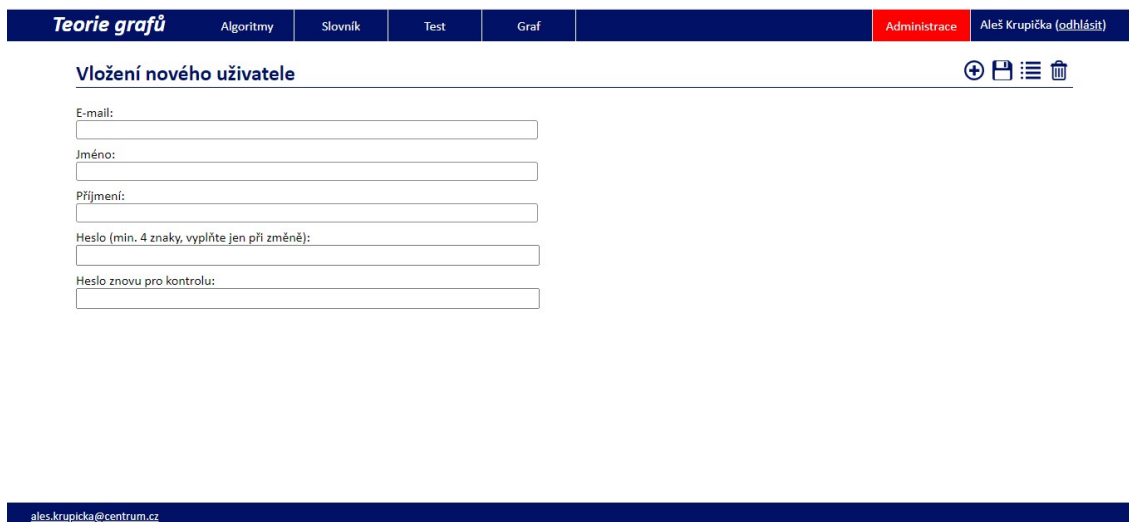
Administrátor může editovat jednotlivé uživatele. V příslušné sekci lze zobrazit jejich přehled, viz obrázek 49.



Obrázek 49 Přehled uživatelů (zdroj: vlastní zpracování)

V editačním okně lze měnit vlastnosti uživatelů, viz obrázek 50:

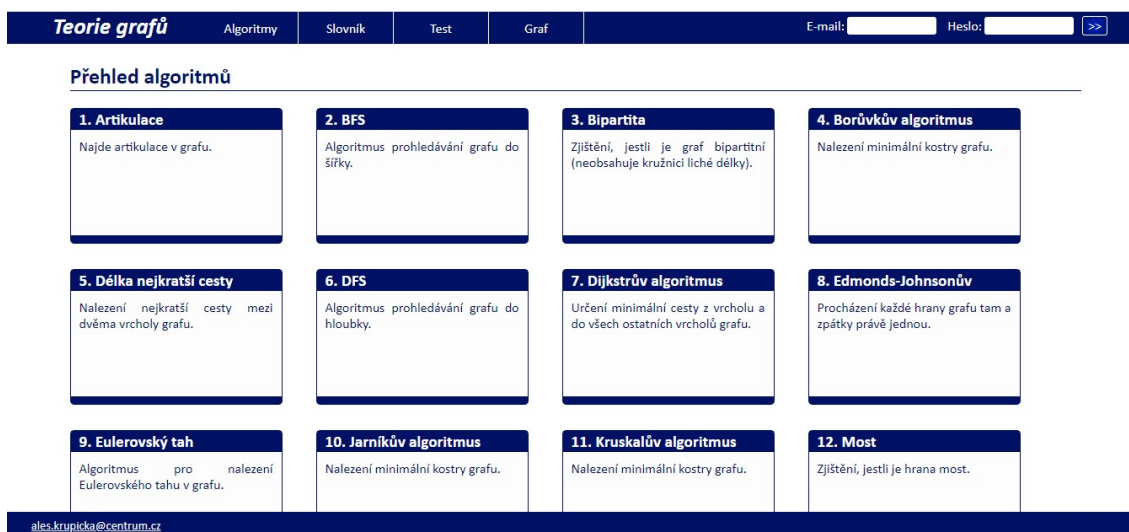
- jméno,
- příjmení,
- e-mail (slouží k přihlášení),
- heslo



Obrázek 50 Editace uživatelů (zdroj: vlastní zpracování)

4.7 Ovládání studentské části

Studentské sekce webové aplikace je přístupná bez přihlašovacích údajů. V této kapitole bude následovat stručný popis a ovládání. Titulní strana obsahuje přehled algoritmů, které aplikace nabízí a student si je může osvojit, viz obrázek 51.



Obrázek 51 Titulní stránka – přehled algoritmů (zdroj: vlastní zpracování)

4.7.1 Algoritmy

Každý algoritmus lze studovat v několika formách. Nabídka je vidět na titulní stránce algoritmu, kde je také první z nich – textový popis algoritmu a jeho principu, viz obrázek 52.

Teorie grafů Algoritmy Slovník Test Graf E-mail: Heslo: >>

BFS Obecné Ukázka Video Cvičení Vyřešit Test

BFS – popis algoritmu

Anglický název
Breadth-first Search

K čemu algoritmus slouží?
Prohledání vrcholů grafu, popř. nalezení konkrétní hodnoty vrcholu.

Časová složitost
» $O(|V| + |E|)$ – každá hrana i vrchol jsou procházeny pouze jednou

Vstup
» souvislý graf $G(V,E)$

Iniciaizace
» libovolný vrchol vložený do fronty FIFO

Postup
» Všem vrcholům grafu nastav příznak *neprohledán*.
» Počátečnímu vrcholu s nastav příznak *prohledán*.
» Vlož počáteční vrchol do fronty FIFO.

ales.krupicka@centrum.cz

Obrázek 52 Textový popis algoritmu (zdroj: vlastní zpracování)

Další možností jsou ukázky, které do systému vložil administrátor. Jejich nabídka je vpravo nahoře příslušné stránky, viz obrázek 53. Uživatel si může ukázky procházet pomocí ovládacích prvků nebo kláves šipek. Průběh algoritmu je slovně popsán a barevně znázorněn v grafu.

Teorie grafů Algoritmy Slovník Test Graf E-mail: Heslo: >>

BFS Obecné **Ukázka** Video Cvičení Vyřešit Test

BFS – ukázka 1 2 3 4 5 6

« < > » ▶ ■

Následuje vrchol F, který je ale prohledán, proto hranu DF vyloučíme.

Neprohledané
H

Prohledané
ABCDEF

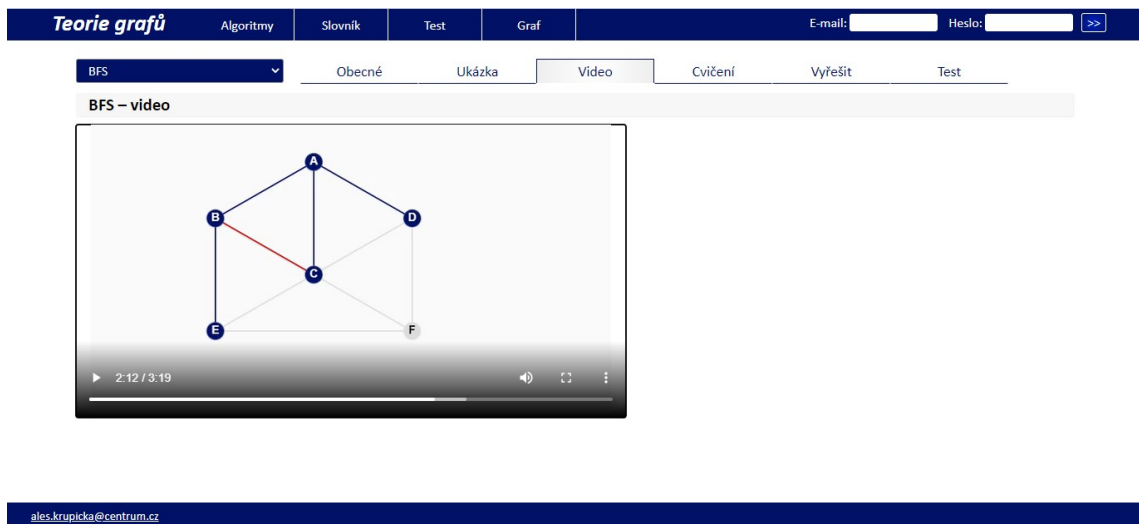
Stromové hrany
AD, AC, AE, BD, CF, CG

Nestromové hrany
CE, DF

ales.krupicka@centrum.cz

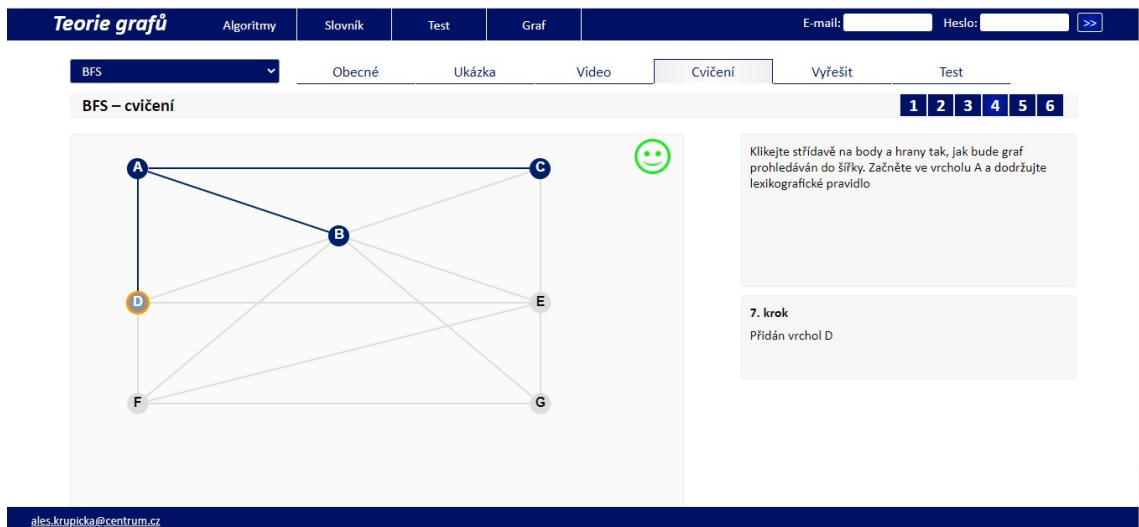
Obrázek 53 Ukázka běhu algoritmu (zdroj: vlastní zpracování)

Další formou demonstrování algoritmů jsou videoukázky. V nich je průběh algoritmu nejen viditelný, ale i slovně okomentovaný, viz obrázek 54.



Obrázek 54 Videouázka algoritmu (zdroj: vlastní zpracování)

V části cvičení si může student osvojené principy procvičit. Nabídka cvičení pro daný algoritmus se nachází vpravo nahoře příslušné stránky. Úkol studenta je klikat na elementy grafu v duchu daného algoritmu podle instrukcí. Aplikace graficky i akusticky hodnotí každou akci studenta, viz obrázek 55.



Obrázek 55 Cvičení algoritmu (zdroj: vlastní zpracování)

V sekci vyřešit může student aplikovat algoritmus na jím zadaný graf (viz 4.7.4). Kromě vyřešení algoritmu jsou v informačním sektoru zobrazovány i relevantní informace, viz obrázek 56.

The screenshot shows the 'Teorie grafů' website interface. At the top, there are navigation tabs: 'Algoritmy', 'Slovník', 'Test', 'Graf'. The 'Algoritmy' tab is selected, and 'BFS' is chosen from a dropdown menu. Below the navigation, there are buttons for 'Obecné', 'Ukázka', 'Video', 'Cvičení', 'Vyřešit', and 'Test'. The main content area is titled 'BFS – řešení' and displays a graph with nodes A, B, C, D, E, F. Node A is highlighted in orange, and edges (A,B), (A,C), and (A,D) are also highlighted in orange. To the right of the graph, a list of steps is shown: 1. Přidán uzel A, 2. Přidána hrana AB, 3. Přidán uzel B, 4. Přidána hrana AC, 5. Přidán uzel C, 6. Přidána hrana AD, 7. Přidán uzel D. At the bottom left of the page, the email 'ales.krupicka@centrum.cz' is visible.

Obrázek 56 Vyřešení algoritmu na vlastním grafu (zdroj: vlastní zpracování)

V části *Test* si student může ověřit faktické znalosti o příslušném algoritmu, pokud jsou k němu přiřazeny otázky, viz obrázek 57.

The screenshot shows the 'Teorie grafů' website interface. At the top, there are navigation tabs: 'Algoritmy', 'Slovník', 'Test', 'Graf'. The 'Test' tab is selected. Below the navigation, there are buttons for 'Obecné', 'Ukázka', 'Video', 'Cvičení', 'Vyřešit', and 'Test'. The main content area displays a test question: '3. Kolik hran tvoří strom prohledávání, který vznikne při prohledávání do šířky grafu, který má 10 vrcholů?'. The options are: a) 9, b) 10, c) 11, d) Nelze jednoznačně určit. The option 'b) 10' is highlighted in red. Below the question, there is a button labeled '» Vyhodnocení »'. At the bottom of the page, a progress bar shows 66.67% completion. At the bottom left of the page, the email 'ales.krupicka@centrum.cz' is visible.

Obrázek 57 Test pro konkrétní algoritmus (zdroj: vlastní zpracování)

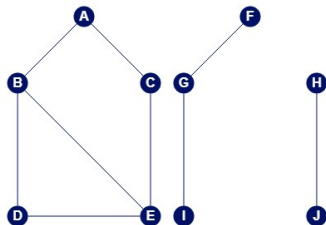
4.7.2 Slovník

Sekce Slovník nabízí přehled důležitých pojmů teorie grafů. Student si zde může vyhledat jemu neznámý pojem. Ten je vysvětlen textově, a pokud je to možné, tak i ilustrativně, viz obrázek 58.

souvislý graf

TEST DEFINICE

Řekneme, že graf G je **souvislý**, jestliže pro každé jeho dva vrcholy x a y existuje v G cesta z x do y .
Souvislý graf obsahuje jedinou komponentu (je tvořena celým grafem).



Graf ABCDE je souvislý, graf FGHI je nesouvislý.

Obrázek 58 Vysvětlení pojmu ve slovníku (zdroj: vlastní zpracování)

K některým pojmům se vztahují testové otázky, a je tedy možné si udělat test. Principiálně je stejný jako v kapitole 4.6.5.

4.7.3 Test

Další sekci je *Test*, kde je možné hlubší ověření znalostí. Při spuštění jsou v náhodném pořadí položeny všechny otázky. V průběhu testu je graficky znázorňována úspěšnost a na konci je test vyhodnocen, viz obrázek 59.

Test

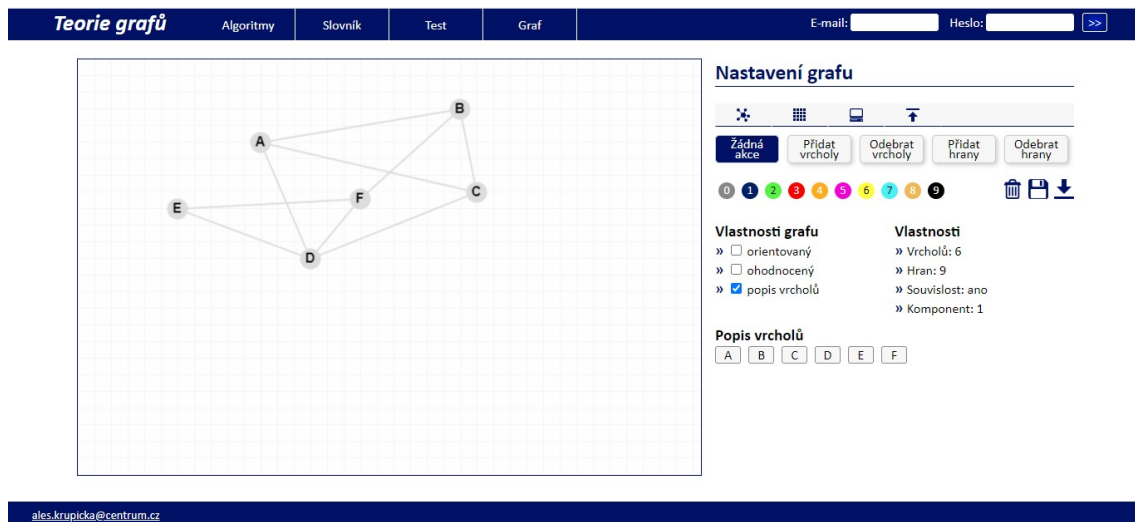
Spustit test



Obrázek 59 Úvodní obrazovka testu (zdroj: vlastní zpracování)

4.7.4 Graf

V sekci Graf může uživatel vložit a editovat graf podle své potřeby. Na něj může následně aplikovat relevantní algoritmy. Kromě toho může graf uložit do souboru, popř. ho ze souboru opětovně načíst do aplikace, viz obrázek 60.



Obrázek 60 Editace testu studentem (zdroj: vlastní zpracování)

Editační stránka umožňuje:

- přidávat a odebírat vrcholy a hrany,
- přemísťovat vrcholy,
- obarvovat vrcholy a hrany,
- přepínat orientaci grafu,
- přepínat a nastavovat ohodnocení hran,
- přepínat popis vrcholů.

5 Testování aplikace

5.1 Podmínky testování

Aplikace byla testována v průběhu výuky předmětu *Seminář z programování* na Gymnáziu Praha 4, Budějovická 680. Testování studenti navštěvovali maturitní ročník čtyřletého, šestiletého nebo osmiletého studia. Věk studentů byl tedy 19 let. Testovanou částí aplikace byly algoritmy pro nalezení minimální kostry v neorientovaném souvislém kladně ohodnoceném grafu.

Jelikož bylo cílem zjistit vliv aplikace na pochopení teorie grafů, bylo důležitým požadavkem, aby se žádný z testovaných studentů dosud s problematikou grafových algoritmů nesetkal. Předchozí nabyté vědomosti (např. absolvované přednášky nebo cvičení) by mohly ovlivnit schopnost studentů vyřešit úlohy bez vlivu aplikace. Proto byli za respondenty zvoleni takoví studenti, kteří splňovali následující požadavky:

- Jejich věk se blížil věku studentů vysoké školy,
- byli přípořně zaměřeni (zvolili si blok přírodovědných předmětů),
- nesetkali se s testovanou problematikou,
- měli zájem se o pro ně neznámý obor a pojmout testování poctivě.

Testování proběhlo v říjnu a listopadu 2019.

5.2 Průběh testování

5.2.1 První fáze – teoretické vysvětlení problematiky, výklad

Studentům byl vysvětlen účel celého experimentu, byli požádáni, aby dodržovali instrukce, a také byli informováni, že se výsledky závěrečného ověřování jejich znalostí nijak nepromítnou do jejich klasifikace. Testování studenti byli rozděleni na 2 skupiny (označené A a B) po deseti. Oběma testovaným skupinám byly vyloženy základní pojmy teorie grafů potřebné pro osvojení postupu pro nalezení minimální kostry grafu [39]. Šlo zejména o termíny vrchol, hrana, graf, orientace hrany, ohodnocení hrany, sled, cesta, tah, kružnice, strom, souvislost grafu, podgraf, kostra grafu, minimální kostra grafu a lexikografické pravidlo. Dále byl studentům vysvětlen praktický případ využití minimální kostry grafu.

Časová dotace této části testování byla tři vyučovací hodiny. Studenti byli požádáni, aby své znalosti nijak nerozšiřovali samostudiem ani nekonzultovali navzájem.

5.2.2 Druhá fáze – studium algoritmů, výklad a samostatná práce

Druhá část testu proběhla v každé skupině rozdílně. Studenti byli informováni, že jejich úkolem je osvojit si znalosti tří algoritmů pro získání minimální kostry grafu:

- Jarníkův algoritmus,
- Kruskalův algoritmus,
- Borůvkův algoritmus.

Dále jim bylo sděleno, že ověření jejich znalostí bude probíhat v papírové podobě a že se bude skládat z vyřešení příkladu, ale také z teoretických otázek na činnost algoritmu.

Skupina A absolvovala další výklad, kde byly studentům tyto algoritmy popsány. Pro každý z algoritmů byl nejdříve popsán a vysvětlen postup nalezení minimální kostry. Poté byl postup demonstrován na dvou příkladech pro každý algoritmus.

Studentům skupiny B byl na začátku druhé fáze testování stručně připomenut pojem minimální kostra. Následně dostali studenti k prozkoumání relevantní část testované aplikace. Studenti měli k dispozici pro každý testovaný algoritmus šest ukázek a šest cvičení. Byli požádáni, aby výhradně s využitím aplikace zjistili, jak jednotlivé algoritmy fungují, a to do té míry, aby dokázali:

- postup každého algoritmu předvést na předloženém grafu v papírové podobě,
- popsat každý algoritmus a uvést rozdíly.

Ukázky umožňovaly procházet průběh vzniku minimální kostry. Cvičení vyžadovala od studentů akci, při níž se postupným klikáním museli zvolit správný element, a postupně tak tvořit minimální kostru. Správně označené elementy (vrcholy i hrany) vznikající minimální kostry byly barevně odlišeny. Při špatné volbě elementu byl student o chybě vizuálně i akusticky informován. Počet pokusů byl neomezený. Studentům skupiny B nebyl žádnou formou vysvětlen postup jednotlivých algoritmů. Jejich úkolem bylo osvojit si požadované znalosti pouze prostřednictvím

části aplikace, kterou měli plně k dispozici. Studentům nebyly poskytovány žádné další informace, pouze zodpovězeno několik dotazů pro připomenutí znalostí z druhé části. Také jim bylo sděleno, že bude požadován konkrétní algoritmus.

Časová dotace druhé fáze byla tři vyučovací hodiny a na jejím konci byli studenti opět požádáni, aby své znalosti schopnosti žádným způsobem nerozšiřovali. Byla také deaktivována aplikace.

5.2.3 Třetí fáze – ověření získaných znalostí

Studentům obou skupin byl předložen shodný test. Bylo jim znovu zdůrazněno, aby se žádným způsobem nepokoušeli podvádět, že výsledky nebudou mít vliv na jejich klasifikaci. Test obsahoval následující otázky a úkoly:

Kolik vrcholů obsahuje kostra grafu?

- a) minimálně tři vrcholy,
- b) minimálně polovinu všech vrcholů grafu,
- c) všechny vrcholy grafu,
- d) nelze jednoznačně určit.

Kolik hran obsahuje kostra grafu?

- a) minimálně tři hrany,
- b) minimálně polovinu všech hran grafu,
- c) o jednu hranu méně, než počet vrcholů,
- d) nelze jednoznačně určit.

Který z algoritmů nalezne nejvýhodnější minimální kostru grafu?

- a) Jarníkův algoritmus,
- b) Borůvkův algoritmus,
- c) Kruskalův algoritmus,
- d) všechny stejně, minimální kostra je unikátní,
- e) nelze jednoznačně určit, záleží na grafu.

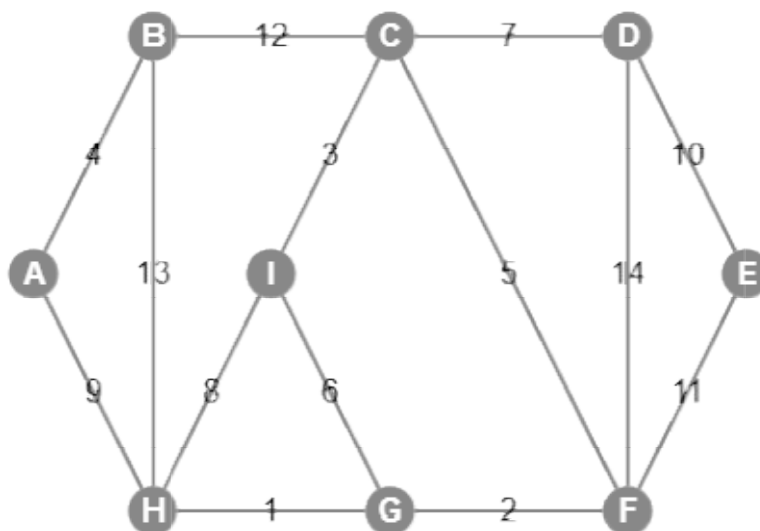
Uveďte tři příklady praktického využití minimální kostry grafu, které nebyly uvedeny při výkladu.

U následujícího grafu nalezněte minimální kostru využitím Jarníkova algoritmu. Zapište pořadí přidávaných hran (např. AB, ED, AC atd.)

U následujícího grafu nalezněte minimální kostru využitím Kruskalova algoritmu. Minimální kostru graficky zvýrazněte a zapište sled přidávaných hran (např. AB, ED, AC atd.).

U následujícího grafu nalezněte minimální kostru využitím Borůvkova algoritmu. Začněte ve vrcholu A a dodržujte lexikografické pravidlo. Minimální kostru graficky zvýrazněte a zapište sled přidávaných hran (např. AB, ED, AC atd.).

U všech tří úloh byl znázorněn graf, viz obrázek 61. Studenti měli k dispozici i náhradní listy s grafem pro případ chybného postupu. Časová dotace pro vyhotovení byla jedna vyučovací hodina.



Obrázek 61 Zadání pro hledání minimální kostry (zdroj: vlastní zpracování)

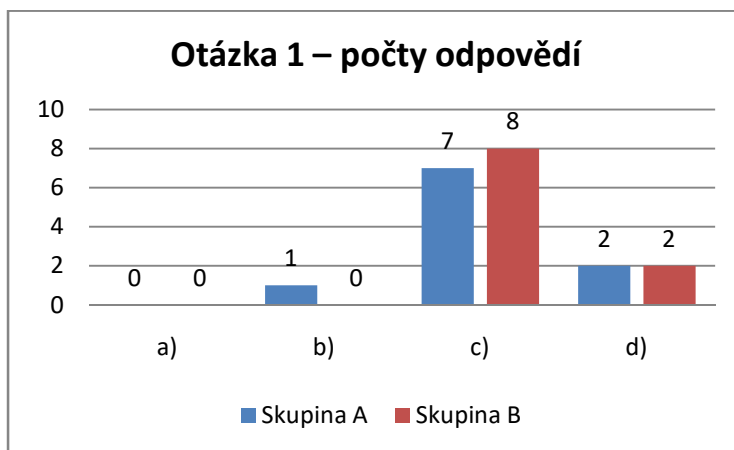
5.2.4 Výsledky testu

5.2.4.1 Úloha 1

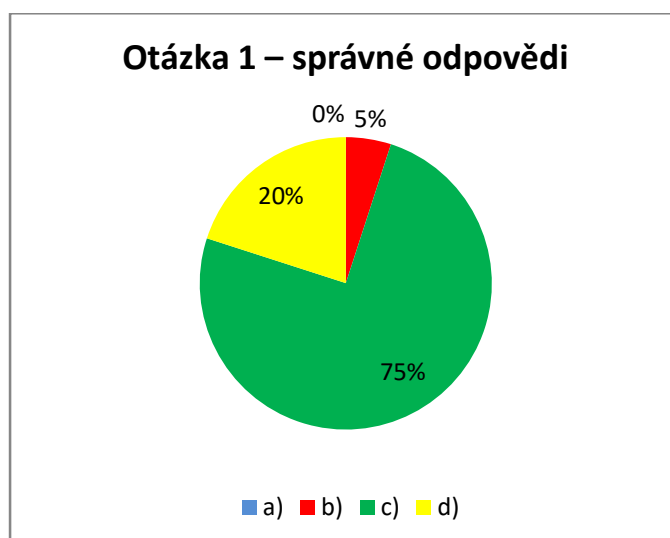
Kolik vrcholů obsahuje kostra grafu?

- a) minimálně tři vrcholy,
- b) minimálně polovinu všech vrcholů grafu,
- c) **všechny vrcholy grafu,**
- d) nelze jednoznačně určit.

Počty a poměrné rozložení správných a chybných odpovědí jsou patrné z grafu 1 a grafu 2.



Graf 1 Otázka 1, počty odpovědí
(zdroj: vlastní zpracování, nástroj MS Excel)



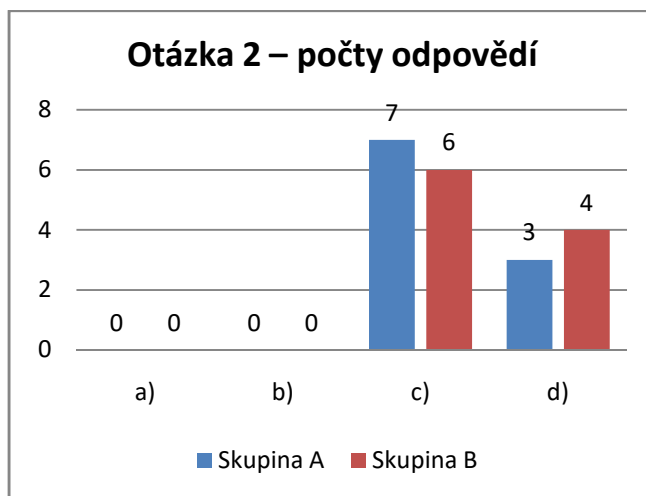
Graf 2 Otázka 1, procento správných a chybných odpovědí
(zdroj: vlastní zpracování, nástroj MS Excel)

5.2.4.2 Úloha 2

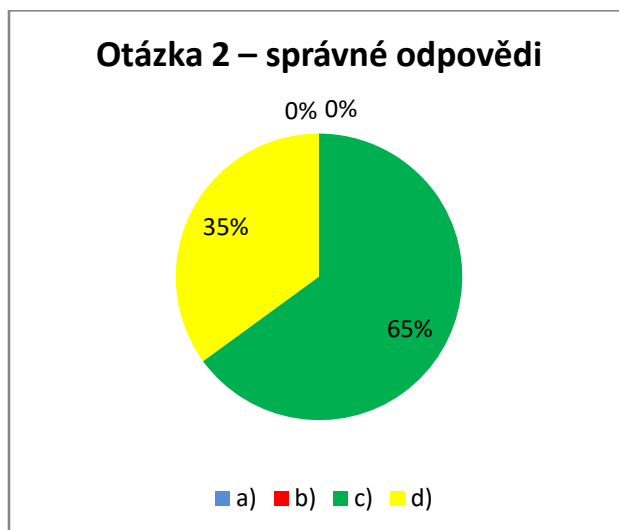
Kolik hran obsahuje kostra grafu?

- a) minimálně tři hrany,
- b) minimálně polovinu všech hran grafu,
- c) **o jednu hranu méně, než počet vrcholů,**
- d) nelze jednoznačně určit.

Počty a poměrné rozložení správných a chybných odpovědí jsou patrné z grafu 3 a grafu 4.



Graf 3 Otázka 2, počty odpovědí
(zdroj: vlastní zpracování, nástroj MS Excel)



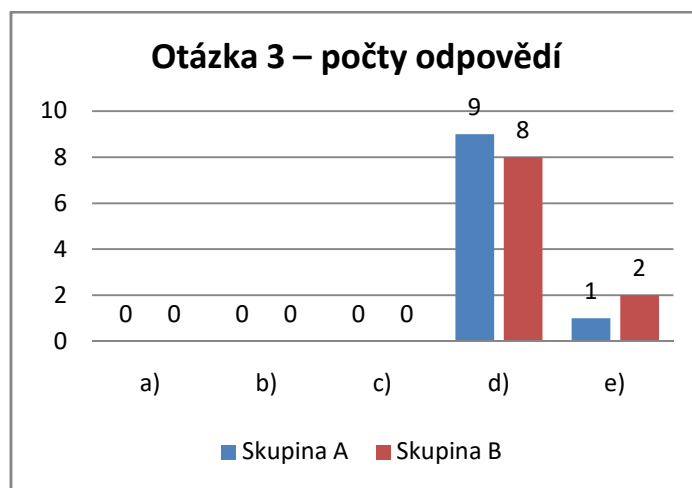
Graf 4 Otázka 2, procento správných a chybných odpovědí
(zdroj: vlastní zpracování, nástroj MS Excel)

5.2.4.3 Úloha 3

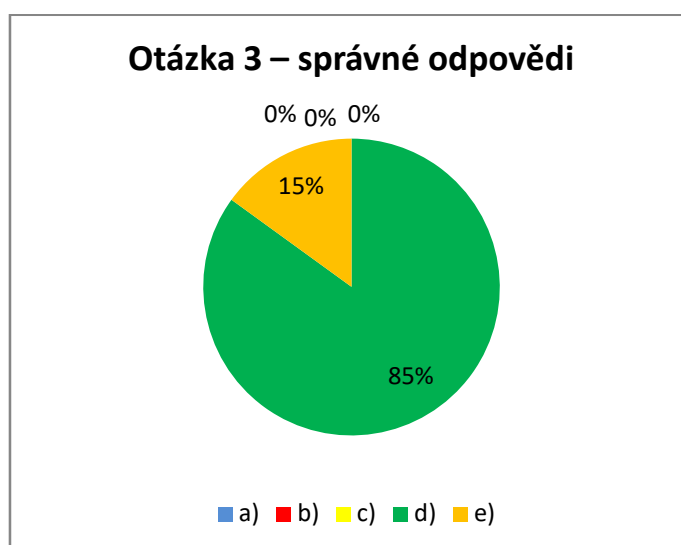
Který z algoritmů nalezne nejvýhodnější minimální kostru grafu?

- a) Jarníkův algoritmus,
- b) Borůvkův algoritmus,
- c) Kruskalův algoritmus,
- d) **všechny stejně, minimální kostra je unikátní,**
- e) nelze jednoznačně určit, záleží na grafu.

Počty a poměrné rozložení správných a chybných odpovědí jsou patrné z grafu 5 a grafu 6.



Graf 5 Otázka 3, počty odpovědí
(zdroj: vlastní zpracování, nástroj MS Excel)

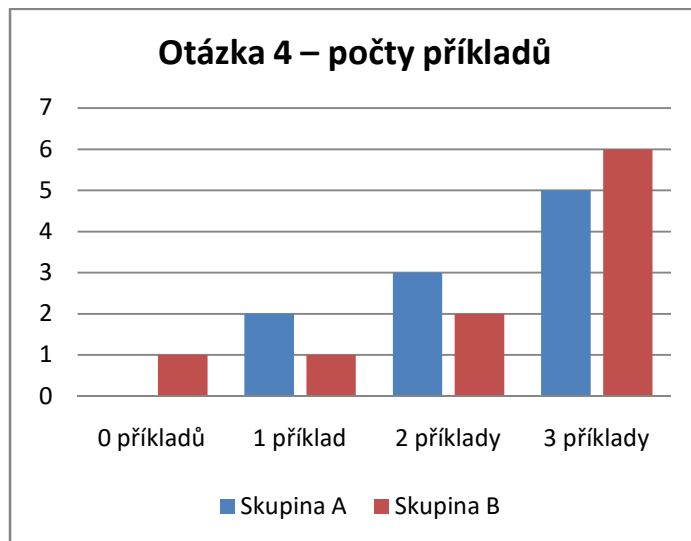


Graf 6 Otázka 3, procento správných a chybných odpovědí
(zdroj: vlastní zpracování, nástroj MS Excel)

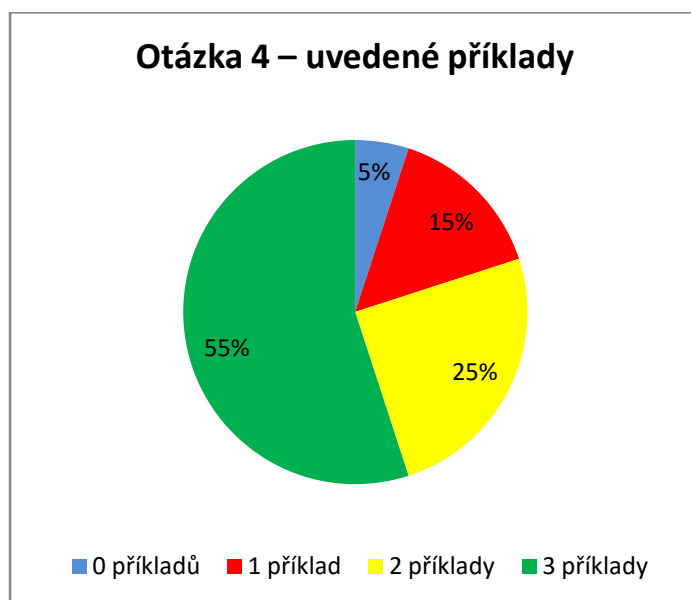
5.2.4.4 Úloha 4

Uved'te tři příklady praktického využití minimální kostry grafu, které nebyly uvedeny při výkladu.

Počty a poměrné rozložení uvedeného počtu příkladů jsou patrné z grafu 7 a grafu 8.



Graf 7 Otázka 4, počty uvedených příkladů
(zdroj: vlastní zpracování, nástroj MS Excel)

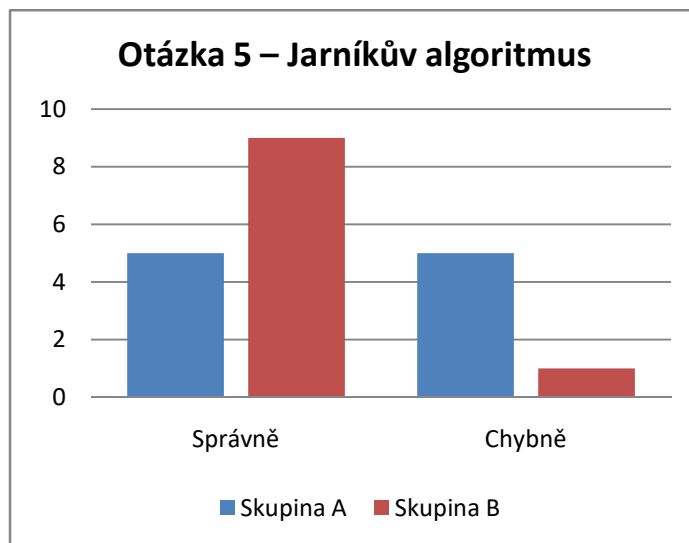


Graf 8 Otázka 4, procento uvedených příkladů
(zdroj: vlastní zpracování, nástroj MS Excel)

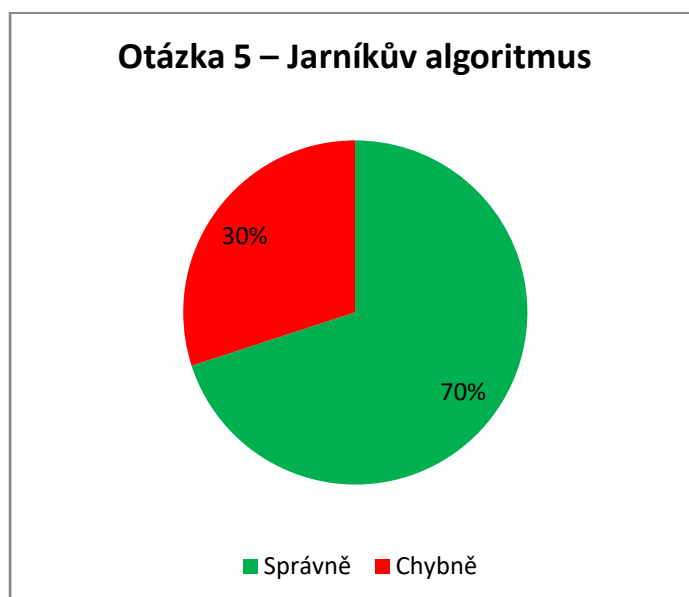
5.2.4.5 Úloha 5

U následujícího grafu naleznete minimální kostru využitím Jarníkova algoritmu. Začněte od vrcholu A a dodržujte lexikografické pravidlo. Minimální kostru graficky zvýrazněte a zapište sled přidávaných hran (např. AB, ED, AC atd.).

Počty a poměrné rozložení správných a chybných řešení jsou patrné z grafu 9 a grafu 10.



Graf 9 Otázka 5, Jarníkův algoritmus, počty správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel)

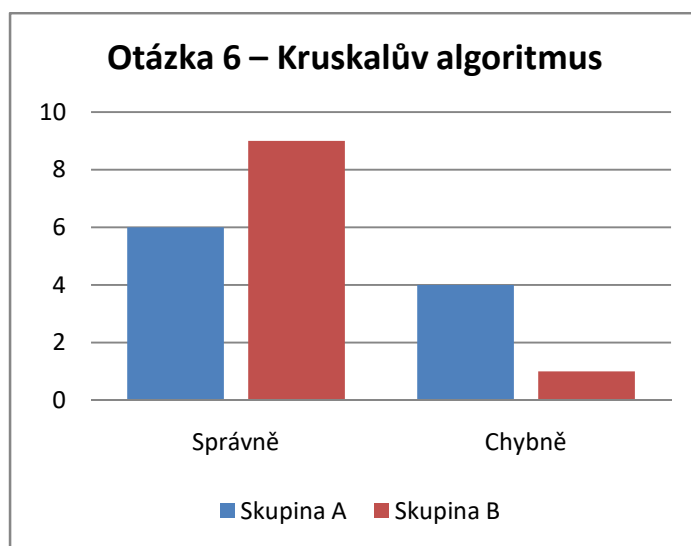


Graf 10 Otázka 5, Jarníkův algoritmus, procento správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel)

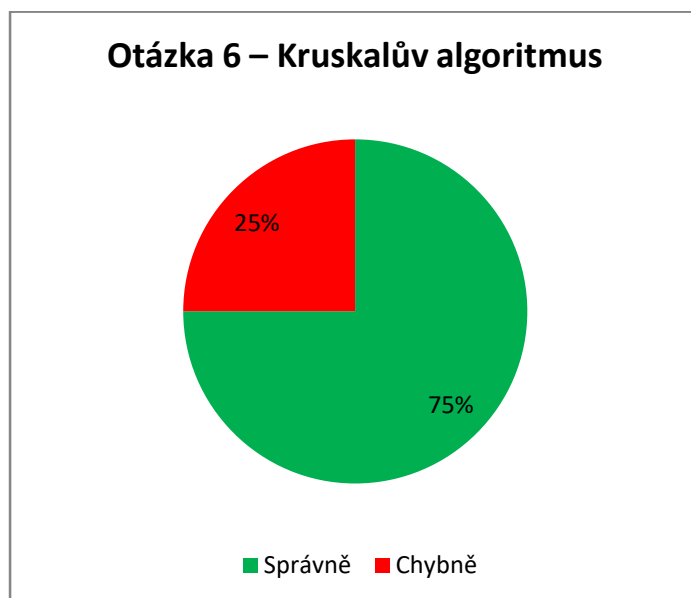
5.2.4.6 Úloha 6

U následujícího grafu naleznete minimální kostru využitím Kruskalova algoritmu. Minimální kostru graficky zvýrazněte a запиšte sled přidávaných hran (např. AB, ED, AC atd.).

Počty a poměrné rozložení správných a chybných odpovědí jsou patrné z grafu 11 a grafu 12.



Graf 11 Otázka 6, Kruskalův algoritmus, počty správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel)

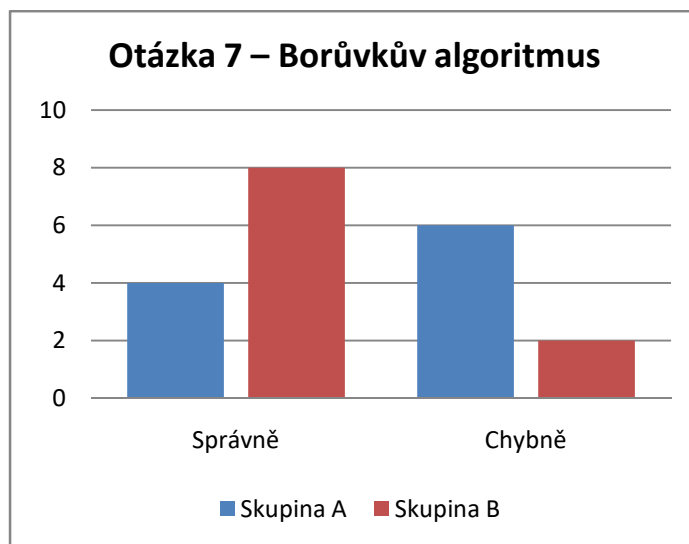


Graf 12 Otázka 6, Kruskalův algoritmus, procento správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel)

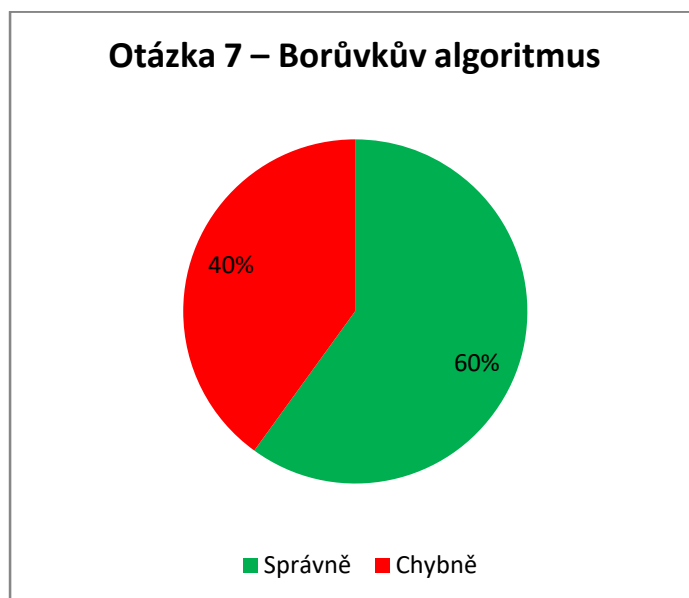
5.2.4.7 Úloha 7

U následujícího grafu naleznete minimální kostru využitím Borůvkova algoritmu. Začněte ve vrcholu A a dodržujte lexikografické pravidlo. Minimální kostru graficky zvýrazněte a zapište sled přidávaných hran (např. AB, ED, AC atd.).

Počty a poměrné rozložení správných a chybných odpovědí jsou patrné z grafu 13 a grafu 14.



Graf 13 Otázka 7, Borůvkův algoritmus, počty správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel)



Graf 14 Otázka 7, Borůvkův algoritmus, procento správných a chybných řešení (zdroj: vlastní zpracování, nástroj MS Excel)

5.2.4.8 Shrnutí a porovnání výsledků ověřování

V následující tabulce 25 jsou přehledně shrnuty výsledky celého testování.

Tabulka 25 Shrnutí výsledků testování (zdroj: vlastní zpracování, nástroj MS Excel)

Výsledky testování		Otázka 1		Otázka 2		Otázka 3		Otázka 5		Otázka 6		Otázka 7	
		Kolik vrcholů		Kolik hran		Výhodnost		Jarník		Kruskal		Borůvka	
		Počet	Poměr	Počet	Poměr	Počet	Poměr	Počet	Poměr	Počet	Poměr	Počet	Poměr
Skupina A	správně	7	70 %	7	70 %	7	70 %	7	70 %	7	70 %	7	70 %
	chybně	3	30 %	3	30 %	3	30 %	3	30 %	3	30 %	3	30 %
	celkem	10	100 %	10	100 %	10	100 %	10	100 %	10	100 %	10	100 %
Skupina B	správně	8	80 %	8	80 %	8	80 %	8	80 %	8	80 %	8	80 %
	chybně	2	20 %	2	20 %	2	20 %	2	20 %	2	20 %	2	20 %
	celkem	10	100 %	10	100 %	10	100 %	10	100 %	10	100 %	10	100 %
Celkem	správně	15	75 %	15	75 %	15	75 %	15	75 %	15	75 %	15	75 %
	chybně	5	25 %	5	25 %	5	25 %	5	25 %	5	25 %	5	25 %
	celkem	20	100 %	20	100 %	20	100 %	20	100 %	20	100 %	20	100 %

5.2.5 Vyhodnocení testování

Ověřování získaných znalostí a dovedností studentů lze rozdělit na dvě části. Správné odpovědi na první čtyři otázky zazněly v průběhu výkladu, popř. je bylo možné z výkladu vyvodit. Obě skupiny absolvovaly shodný výklad. Z výsledků je zřetelně vidět, že se výsledky v obou skupinách nijak zásadně neliší. Počty správných odpovědí jsou podobné.

Výsledky studentů obou skupin ve zbývajících třech úkolech se lišily podle toho, jakou formou postup algoritmů hledání minimální kostry studovali. Studenti skupiny A vyslechly výklad, ale praktickou zkušenost měli pouze se dvěma příklady, z nichž jeden řešili samostatně.

Studenti skupiny B stejný čas využili samostatným opakovaným praktickým sledováním ukázek a nacvičováním správného postupu. Výsledky ukázaly, že pro praktické ovládnutí uvedených algoritmů je tato metoda významně efektivnější.

6 Závěry a doporučení

Autor zvolil téma teorie grafů pro svou diplomovou práci z několika důvodů. V první řadě ho zaujal obor jako takový. Práce s grafy a grafové algoritmy v sobě snoubí matematickou přesnost, možnost dobré představitivosti, programátorskou realizovatelnost, praktické využití a také jistou formu elegance. Dalším důvodem byl subjektivní pocit potřeby podobného výukového nástroje během vlastního studia. A v neposlední řadě lidský přístup vedoucí diplomové práce, o němž se autor přesvědčil už v průběhu výuky.

Autor se v průběhu vývoje této práce zdokonalil jak v problematice teorie grafů a v chápání jednotlivých algoritmů, tak v programování. Zároveň se díky práci naučil posoudit možnosti co největší výukové přínosnosti, aby jí mohl přizpůsobit funkce aplikace. V průběhu programování se autor naučil ovládat a implementovat funkce několika knihoven a také skloubit fungování různých programátorských nástrojů. Dalším subjektivním přínosem pro autora byla inspirace zařadit téma teorie grafů do volitelných předmětů na gymnáziu, kde vyučuje – seznámit se základy tohoto oboru studenty a naprogramovat s nimi některé algoritmy.

Stěžejním cílem diplomové práce bylo navržení a vytvoření aplikace pro podporu výuky předmětů DMO, dále sestavení databáze ukázkových příkladů a cvičení. Vytvoření aplikace předcházelo nastudování a teoretické zpracování grafových algoritmů. Aplikace byla vytvořena tak, jak bylo stanoveno. Také byla sestavena sada ukázkových příkladů, kterou je možné nadále uživatelsky rozšiřovat. Dále byla popsána struktura a ovládání aplikace, vytvořena elektronická uživatelská příručka, aplikace byla otestována

Cíle této diplomové práce byly splněny. Vznikla webová aplikace, která usnadní proces učení vyučujícím i studentům. Možnosti aplikace navíc nejsou uzavřeny jejím současným stavem, její charakter ji umožňuje nadále rozšiřovat uživateli (bez dalšího zásahu autora) podle aktuálních výukových potřeb na FIM UHK. Přesto by se autor rád na dalším vývoji aplikace podílel.

7 Seznam použité literatury

1. **Šeda, M.** Teorie grafů. *VUT v Brně*. [Online] 2003. [Citace: 15. 8 2020.]
2. **Matoušek, J. a Nešetřil, J.** *Kapitoly z diskrétní matematiky*. Praha : Karolinum, 2007.
3. **Opava, Z.** *Matematika kolem nás*. Praha : Albatros, 1989.
4. **Biggs, N.** *Discrete Mathematics*. Oxford : Oxford University Press, 2002.
5. **Šišma, P.** *Teorie grafů 1736-1963*. Praha : Prometheus, 1997.
6. **Lučištník, P.** *Interaktivní mapa BTS*. [Online] [Citace: 15. 8 2020.]
<https://www.gsmweb.cz/mapa/>.
7. **Dylan, M.** Dotted Polylines with Google Maps SDK for iOS. *Door2door Engineering*. [Online] 10. 4 2018. [Citace: 15. 8 2020.]
<https://engineering.door2door.io/dotted-polylines-with-google-maps-sdk-for-ios-7b21786a0645>.
8. **Hnátek, V.** Chytrá matematika z Valmezu šetří peníze v logistice. *Lupa*. [Online] Internet Info, s.r.o., 1. 7 2020. [Citace: 15. 8 2020.] <https://www.lupa.cz/pr-clanky/chytra-matematika-z-valmezu-setri-penize-v-logistice-1/>.
9. **Milková, E.** Teorie grafů a grafové algoritmy. *Informační systém Masarykovy univerzity*. [Online] [Citace: 13. 8 2020.]
https://is.muni.cz/el/1441/podzim2016/MA2BP_PDM1/grafy-milkova-hradec-2013.txt.
10. **Demel, J.** *Grafy a jejich aplikace*. Praha : Academica, 2002.
11. **Dannhoferová J., Foltýnek T.** Eulerovské a hamiltonovské grafy. *Teorie grafů*. [Online] Mendelova univerzita v Brně. [Citace: 18. červen 2018.]
https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=23498.
12. **Šitina, J.** *Grafové algoritmy a jejich využití. Diplomová práce*. Hradec Králové : autor neznámý, 2010.
13. **Šťastná, M.** *Vizuální podpora výuky předmětů zabývajících se teorií grafů a grafovými algoritmy. Diplomová práce*. Hradec Králové : autor neznámý, 2017.
14. **Hübner, A.** *Aplikace pro podporu výuky předmětu Diskrétní matematika. Bakalářská práce*. Hradec Králové : Univerzita Hradec Králové, 2014.
15. **Herman, V.** *Barvení grafů. 2015. Diplomová práce*. Hradec Králové : Univerzita Hradec Králové, Fakulta informatiky a managementu., 2015.

16. **Krátký, R.** *Hamiltonovské grafy. Diplomová práce.* Hradec Králové : Univerzita Hradec Králové, Fakulta informatiky a managementu., 2020.
17. **Skořepa, T.** *Vizuální podpora výuky předmětů zabývajících se teorií grafů a grafovými algoritmy. Diplomová práce.* Hradec Králové : autor neznámý, 2018.
18. **Bartoš, M.** *Teorie grafů – vizuální podpora. Bakalářská práce.* Hradec Králové : Univerzita Hradec Králové, Fakulta informatiky a managementu, 2012.
19. **Braha, V.** *Vizuální podpora teorie grafů.* Hradec Králové : Univerzita Hradec Králové, Fakulta informatiky a managementu.
20. **Robertson, N., a další.** *The Four-Colour Theorem, Journal of Combinatorial Theory.* 1997.
21. **Richta, K.** *Datové struktury a algoritmy.* Praha : Katedra počítačů FEL ČVUT Praha, 2018.
22. **Sipser, M.** *Introduction to the theory of computation. 2nd ed.* Boston : International ed. Boston, 2006, 2006. 0-619-21764-2.
23. **Turzík, D. a Pavlíková, P.** *Diskrétní matematika.* Praha : VŠCHT, 2007.
24. **Šedivý, J.** *O konečných grafech a jejich použití.* 1966. stránky 496-509.
25. Prohledávání grafu. *Mendelova univerzita v Brně.* [Online] [Citace: 16. 8 2020.] https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=19950.
26. Minimální kostra grafu. *Univerzitní informační systém Mendelovy univerzity v Brně.* [Online] [Citace: 12. 8 2020.] https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=9737.
27. **Anderson, I.** *A First Course in Discrete Mathematics.* London : Springer, 2001.
28. VŠB Technická univerzita Ostrava. [Online] [Citace: 13. 8 2020.] http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/uvod_do_teorie_grafu.pdf.
29. **Neckář, J.** *Algoritmy.net. Dijkstrův algoritmus.* [Online] [Citace: 13. 8 2020.] <https://www.algoritmy.net/article/5108/Dijkstruv-algoritmus>.
30. Hledání nejkratších cest grafem. *Vysoké učení technické v Brně.* [Online] [Citace: 13. 8 2020.] https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=115691.
31. **Matějka, P.** *Algoritmy pro generování a řešení bludišť (Diplomová práce).* *Masarykova univerzita.* [Online] 2012. [Citace: 16. červen 2018.] https://is.muni.cz/th/zvinp/DP_-_matejka.pdf.

32. **Šeda, M.** Teorie grafů. *VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA STROJNÍHO INŽENÝRSTVÍ*. [Online] 2003. [Citace: 15. 8 2020.]
33. **Ryjáček, Z.** Teorie grafů a diskrétní optimalizace 1. [Online] 16. 9 2007. [Citace: 20. 8 2020.] <ftp://zcu.arcao.com/zcu/kma/tgd1/TGD1-skripta.pdf>.
34. **Kolář, J.** *Teoretická informatika*. Praha : ČVUT, 2009.
35. *Cytoscape.js*. [Online] [Citace: 15. 8 2020.] <http://js.cytoscape.org>.
36. *jQuery*. [Online] [Citace: 15. 8 2020.] <https://jquery.com>.
37. *PHP: Hypertext Preprocessor*. [Online] [Citace: 15. 8 2020.] <https://www.php.net>.
38. *MySQL*. [Online] [Citace: 15. 8 2020.] <https://www.mysql.com/>.
39. **Jirovský, L.** *Vybrané problémy z teorie grafů ve výuce na střední škole*. Praha : MFF UK, 2008.

Příloha 1 – uživatelský manuál pro administrátora

1 Úvod

Aplikace teoriegrafu.cz je výuková aplikace, primárně určená pro studenty Fakulty informatiky a managementu Univerzity Hradec Králové. Aplikace obsahuje část administrátorskou a uživatelskou. Administrátorská část je určena pro vytvoření obsahu, uživatelská část je výuková. Tato uživatelská příručka obsahuje popis ovládání administrátorské části.

1.1 Sekce aplikace

Webová aplikace teoriegrafu.cz obsahuje následující tematicky oddělené sekce, jejichž ovládání bude nadále popsáno.

- Algoritmy
- Grafy
- Ukázky
- Cvičení
- Test
- Slovník
- Barvy
- Uživatelé

1.2 Přihlášení

Využívání administrátorské části aplikace je podmíněno přihlášením do systému prostřednictvím formuláře v pravém horním rohu obrazovky. Není možné se do aplikace individuálně registrovat, přístupové údaje musí být vytvořeny jiným administrátorem.



Obr. 1 Přihlašovací formulář

Po úspěšném přihlášení administrátor vidí na stejném místě své jméno a příjmení. Je zde možnost se opět ze systému odhlásit.



Obr. 2 Výpis přihlášeného

Kromě toho je zpřístupněna položka v menu s popisem *Administrace*, které odkazuje do administrátorské části aplikace.



Obr. 3 Odkaz na administraci

Administrátorská část aplikace je uvedena rozcestníkem, odkazující na jednotlivé sekce. Jejich podrobný popis bude následovat.

Algoritmy <ul style="list-style-type: none">» Přehled a úprava algoritmů» Vložení nového algoritmu	Grafy <ul style="list-style-type: none">» Přehled a úprava grafů» Vložení nového grafu	Ukázky <ul style="list-style-type: none">» Přehled a úprava ukázek
Cvičení <ul style="list-style-type: none">» Přehled a úprava cvičení	Test <ul style="list-style-type: none">» Otázky s možnostmi» Otázky otevřené» Vygenerování PDF	Barvy <ul style="list-style-type: none">» Přehled a úprava barev
Slovník <ul style="list-style-type: none">» Přehled a úprava pojmů» Přidat pojem	Uživatelé <ul style="list-style-type: none">» Přehled a úprava uživatelů» Přidat uživatele	

Obr. 4 Rozcestník administrace

2 Algoritmy

2.1 Přehled algoritmů

V sekci algoritmů je možné editovat jednotlivé algoritmy a jejich vlastnosti. Na titulní stránce sekce je přehled algoritmů se stručným přehledem informací a s odkazy na editační stránky.

Algoritmy (18)

#	Název	Uk.	Cv.	Vid.	Kód	Ex.	Veř.	
1.	Artikulace	0	0					
2.	BFS	10	6					
3.	Bipartita	4	2					
4.	Borůvkův algoritmus	10	8					
5.	Délka nejkratší cesty	4	3					
6.	DFS	7	6					

Obr. 5 Výpis algoritmů

Ovládací prvky tabulky

- Uk.** počet ukázek u algoritmu
- Cv.** počet cvičení u algoritmu
- editace videa
- editace kódu
- informace o existenci kódu
- přepínání zveřejnění algoritmu
- smazání algoritmu

Na každé stránce se také v pravém horním rohu nacházejí ovládací prvky (ikonky). Tyto ikonky budou popsány pro každou stránku. Pro stránku přehledu algoritmů jde o jedinou ikonku.

#	Název	Uk.	Cv.	Vid.	Kód	Ex.	Ver.
1.	Artikulace	0	0	⊙	↔	✓	YPS
2.	BFS	10	6	⊙	↔	✓	YPS
3.	Bipartita	4	2	⊙	↔	✓	YPS
4.	Borůvkův algoritmus	10	8	⊙	↔	✓	YPS
5.	Délka nejkratší cesty	4	3	⊙	↔	✓	YPS
6.	DFS	7	6	⊙	↔	✓	YPS
7.	Dijkstrův algoritmus	2	0	⊙	↔	✓	YPS
8.	Edmonds-Johnsonův	3	2	⊙	↔	✓	YPS
9.	Eulerovský tah	4	3	⊙	↔	✓	YPS
10.	Jarníkův algoritmus	9	7	⊙	↔	✓	YPS
11.	Kruskalův algoritmus	9	8	⊙	↔	✓	YPS
12.	Most	3	2	⊙	↔	✓	YPS
13.	Počet komponent	3	2	⊙	↔	✓	YPS
14.	Souvislost grafu	3	2	⊙	↔	✓	YPS
15.	Tarryho algoritmus	0	0	⊙	↔	✓	YPS
16.	Trémauxův algoritmus	3	2	⊙	↔	✓	YPS
17.	Vrchol na kružnici	3	2	⊙	↔	✓	YPS
18.	Vrcholy v komponentě	5	4	⊙	↔	✓	YPS

Obr. 6 Výpis algoritmů (náhled celé stránky)

Ovládací prvky stránky

vložení nového algoritmu

Kliknutím na název algoritmu dojde k přesměrování na editační formulář daného algoritmu.

2.2 Editace algoritmu

Obr. 7 Formuláře pro editaci algoritmu

U algoritmu lze editovat následující položky:

- **Název** – název algoritmu, řetězec v maximální délce 100 znaků (povinná položka),
- **Perex** – stručný popis algoritmus, text v délce jedné věty,
- **Popis** – obsáhlejší formátovaný popis algoritmu
- **Hodnocení hran v ukázkách** – příznak, který určuje, zda má v ukázkách zůstat u grafů viditelné hodnocení (ocenění) hran.








Dále jsou tu odkazy na editaci ukázek a cvičení s vazbou na příslušný algoritmus (popis viz dále).

Ukázky: [Ukázka 1](#), [Ukázka 2](#), [Ukázka 3](#), [Ukázka 4](#), [Ukázka 5](#), [Ukázka 6](#), [Ukázka 7](#)

Cvičení: [Cvičení 1](#), [Cvičení 2](#), [Cvičení 3](#), [Cvičení 4](#), [Cvičení 5](#), [Cvičení 6](#), [Cvičení 7](#)

Obr. 8 Výpis ukázek a cvičení

Ovládací prvky stránky

-  vložení nového algoritmu
-  uložení změn provedených ve formuláři
-  přehled algoritmů
-  editace kódu
-  editace videa
-  přepínání zveřejnění algoritmu
-  smazání algoritmu (nesmí obsahovat ukázky ani cvičení)

2.3 Editace videa

Kliknutím na příslušnou ikonku je možné načíst video pro příslušný algoritmus. Načtené video je následně možné na stejné stránce smazat pomocí ikonky koše.

Vložte video:
 Soubor nevybrán

Obr. 9 Formulář pro načtení videa

2.4 Editace kódu

Kliknutím na příslušnou ikonku lze editovat kód řešící daný algoritmus.

```
Kód:  
var bfs = cy.elements().bfs('#A', function() {}, false);  
var cesta = bfs.path;  
  
var reset_graf_alg = function() {
```

Obr. 10 Formulář pro editaci kódu

Poznámka: Editací kódu algoritmu je ovlivněn jeho chod a je nutné editaci provádět obezřetně.

3 Grafy

3.1 Přehled grafů

Grafy jsou rozděleny na dvě skupiny.

- a) ukázkové grafy,
- b) ostatní grafy

K ukázkovým grafům se vážou nějaké ukázky anebo cvičení. Ostatní grafy jsou určeny pro další účely, např. pro přípravu na výuku, tvorbu papírových výukových materiálů apod. Grafy lze mezi skupinami libovolně přemísťovat kliknutím na symbol oka.

#	Ukázkové grafy	Uk.	Cv.	Veř.	Uk.	#	Ostatní grafy	Uk.	Cv.	Veř.	Uk.
1.	Ukázky 02 – kostra	16	16			18.	Artikulace	0	0		
2.	Ukázky 03 – kostra, prohledávání	25	25			19.	Artikulace 2	0	0		
3.	Ukázky 06 – prohledávání	4	4			20.	Artikulace txt	0	0		
4.	Ukázky 08 – prohledávání	9	9			21.	Bipartita	0	0		
5.	Ukázky 09 – most	1	1			22.	Bipartita 2	0	0		
6.	Ukázky 10 – most	1	1			23.	Bipartita txt	0	0		
7.	Ukázky 11 – komponenty	16	16			24.	Bipartita ukázka	1	0		
8.	Ukázky 12 – komponenty	1	1			25.	Bloky txt	0	0		

Obr. 11 Přehled grafů

Ovládací prvky tabulky

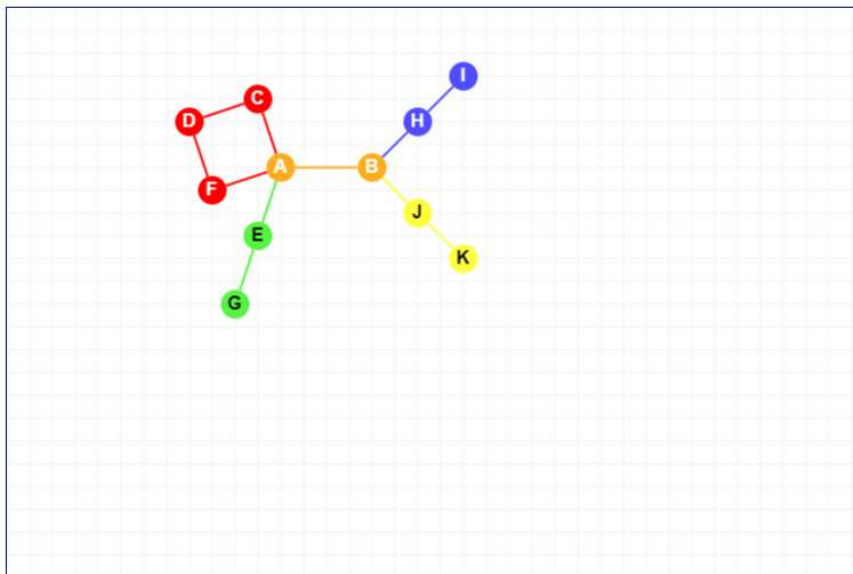
- Uk.** počet ukázek vztahujících se ke grafu
- Cv.** počet cvičení vztahujících se ke grafu
- přepínání zveřejnění grafu
- přepínání mezi oběma skupinami grafů
- smazání grafu

Ovládací prvky stránky

- vložení nového grafu

3.2 Editace grafu

Editační stránka grafu je rozdělena na dvě části. V levé je vizualizace, vpravo ovládací prvky. Pro lepší umístění jednotlivých vrcholů je graf podložen mřížkou.



Obr. 12 Vizualizace grafu při editaci

Editace grafu + 📄 📄 ☰ 🚩 🗑️

1.00 750x500

🔍 📄 ☰ 🗑️ ?

Žádná akce
Přidat vrcholy
Odebrat vrcholy
Přidat hrany
Odebrat hrany

0 1 2 3 4 5 6 7 8 9

Vlastnosti grafu

- » orientovaný
- » ohodnocený
- » popis vrcholů

Vlastnosti

- » Vrcholů: 9
- » Hran: 14
- » Souvislost: ano
- » Komponent: 1

Ohodnocení hran

AB	6	BC	2	CF	8	FI	13	IH	9	HG	7
GD	10	DA	3	DE	12	EF	4	DB	1	BF	14
DH	5	EI	11								



Popis vrcholů

A	B	C	D	E	F	G	H	I
---	---	---	---	---	---	---	---	---

Obr. 13 Ovládací prvky grafu

Ovládací prvky stránky






- + vložení nového grafu (je nutné vložit název),
- 📄 uložení změn grafu,
- 📄 uložit jako (výchozí pro jiný graf),
- ☰ přehled grafů,

-  přepínání zveřejnění algoritmu,
-  smazání algoritmu (nesmí obsahovat ukázky ani cvičení)

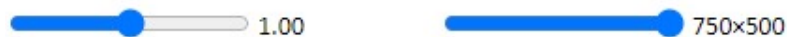
Graf lze editovat pouze tehdy, je-li v nezveřejněném stavu (zašedlý symbol praporeku).

3.3 Možnosti úprav grafu

Úpravy grafu jsou možné v několika oblastech, které charakterizují následující ovládací prvky.

-  základní úpravy – vkládání, mazání a přesuny vrcholů a hran, nastavení barev, editace označení
-  úprava názvu grafu a jeho přiřazení do ukázek a cvičení
-  matice sousednosti
-  export grafu do png
-  nápověda

Kromě toho lze dále měnit velikost pracovní plochy a měřítko grafu.



Obr. 14 Změna velikosti pracovní plochy a měřítko grafu

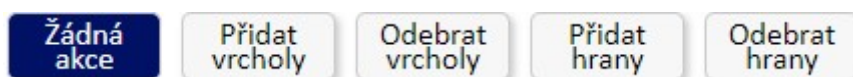
3.4 Základní úpravy

V této části lze editovat základní vlastnosti grafu.

- přidávat a odebírat vrcholy a hrany,
- přemísťovat vrcholy,
- přepínat orientaci grafu,
- přepínat ohodnocení hran a popis vrcholů,
- měnit ohodnocení hran a vrcholů popis vrcholů

3.5 Manipulace s elementy grafu

Veškerou manipulaci s grafem lze provádět přepnutím do příslušného módu:



Obr. 15 Nastavení manipulace s vrcholy a hranami

3.5.1 Přidávání vrcholů

Vrchol lze přidat kliknutím na libovolné místo v grafu. Při tvorbě grafu je vhodné nejdříve přidat všechny vrcholy a následně k nim doplnit hrany.

3.5.2 Odebrání vrcholů

Vrchol je možné smazat kliknutím na něj. Kromě vrcholu jsou odebrány i všechny jeho incidenční hrany.

3.5.3 Přidání hrany

Hranu lze přidat postupným kliknutím na její počáteční a koncový vrchol. I pokud graf není orientovaný, je každá hrana uložena jako orientovaná (jen není orientace zobrazena).

3.5.4 Odebrání hrany

Kliknutím na příslušnou hranu ji lze odebrat.

3.5.5 Přemístění vrcholů

Vrcholy lze kdykoliv libovolně po grafu přemístit pomocí myši při nastavení módu „Žádná akce“. Spolu s vrcholem jsou přesunuty i incidenční hrany. Dalším následným kliknutím na vrchol je tento zarovnán do mřížky.

3.5.6 Přepínání orientace grafu

Vzhledem k tomu, že jsou hrany ukládány jako orientované, je možné snadným přepnutím checkboxu změnit orientaci grafu.

» orientovaný

Obr. 16 Nastavení orientace grafu

3.5.7 Přepínání hodnocení grafu

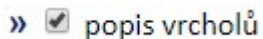
Graf je vždy ukládán jako ohodnocený, hodnocení hran ale mohou být skryta. Zda je graf ohodnocený lze snadno přepnout pomocí checkboxu:

» ohodnocený

Obr. 17 Nastavení ohodnocení grafu

3.5.8 Přepínání popisu vrcholů

Pomocí checkboxu lze také snadno přepínat, zda je viditelný popis vrcholů.



Obr. 18 Nastavení popisu vrcholů

3.5.9 Změna ohodnocení hran

Pokud je nastaveno, že je graf ohodnocený, je možné jednotlivým hranám přiřadit celočíselné ohodnocení.

Ohodnocení hran							
AB	1	BC	5	CD	13	EF	15
FG	7	GH	10	IJ	6	JK	12
KL	3	AE	2	EI	8	BF	20
FJ	16	CG	14	GK	11	DH	9
HL	4						

Obr. 19 Vložení hodnocení hran

3.5.10 Změna popisu vrcholů

Pokud je nastaveno, že jsou viditelné popisy vrcholů, je možné tyto popisy měnit jednoduchým přepsáním na textovou nebo číselnou hodnotu.



Obr. 20 Změna popisu vrcholů grafu

3.5.11 Informace o vlastnostech grafu

Kromě toho jsou zde zobrazeny informace o grafu, konkrétně počet vrcholů, počet hran, souvislost grafu a počet komponent.

Vlastnosti

» Vrcholů: 12

» Hran: 17

» Souvislost: ano

» Komponent: 1

Obr. 21 Vlastnosti grafu

Všechny nastavené vlastnosti je nutné uložit kliknutím na ikonu diskety v pravé horní části okna.

3.6 Úprava názvu grafu a jeho přiřazení do ukázek a cvičení

V druhé části lze nastavit další vlastnosti související s grafem. Jde o:

- název grafu,
- informace,
- přiřazení k algoritmům v ukázkách,
- přiřazení k algoritmům ve cvičeních

The screenshot shows a configuration form for a graph. It is divided into two main sections: 'Název:' and 'Informace:'.
Under 'Název:', there is a text input field containing 'Ukázkový graf'.
Below that is a section 'Přidat ukázkou:' with a dropdown menu set to '-- zvolit algoritmus --'. A blue bar labeled 'Ukázky' is expanded, showing a list with one item: '1. Borůvkův algoritmus' with a trash icon.
Below that is a section 'Přidat cvičení:' with a dropdown menu set to '-- zvolit algoritmus --'. A blue bar labeled 'Cvičení' is expanded, showing a list with two items: '1. Borůvkův algoritmus' and '3. Kruskalův algoritmus', both with trash icons.
At the bottom left is a blue button labeled 'Uložit'.
The 'Informace:' section on the right is a large empty text area.

Obr. 22 Nastavení vlastností grafu, ukázek a cvičení

3.6.1 Název grafu

Název grafu je řetězcová položka o maximální délce 60 znaků, která slouží k identifikaci grafu, hlavně pro potřeby administrátora. Nemusí být unikátní.

3.6.2 Informace

Informace o grafu je textová položka, která slouží pro interní potřeby autora grafu. Pro ostatní uživatele se nikde neobjevuje.

3.6.3 Přiřazení k algoritmům v ukázkách

Má-li graf sloužit pro ukázky, je mu třeba přiřadit příslušný algoritmus. Ten je možné vybrat z rozbalovacího seznamu. Tím vznikne nová ukázka, kterou je ná-

sledně možné editovat v sekci ukázek. Je zde také viditelný seznam všech již vybraných algoritmů (ukázek). Odkazem je možné dostat se přímo na editaci jednotlivé ukázky, popř. ukázku smazat.

3.6.4 Přiřazení k algoritmům ve cvičeních

Podobný mechanismus jako u ukázek funguje u cvičení. Má-li graf sloužit jako cvičení, je mu třeba přiřadit jednotlivý algoritmus. Ten je možné vybrat z rozbalovacího seznamu. Tím vznikne nové cvičení, které je následně možné editovat v sekci cvičení. I zde je viditelný seznam všech již vybraných algoritmů (cvičení). Odkazem je možné dostat se přímo na editaci jednotlivého cvičení, popř. cvičení smazat.

3.7 Matice sousednosti





Ve třetí části je viditelná matice sousednosti grafu, která je generovaná z aktuálně uložené podoby grafu.

	A	B	C	D	E
A	0	5	0	3	8
B	0	0	1	2	0
C	0	0	0	7	0
D	0	0	0	0	0
E	0	4	0	5	0

Obr. 23 Matice sousednosti

3.8 Náповěda

Poslední pátá část zobrazuje nápovědu. Tedy funkci jednotlivých ikoněk, pomocí nichž se graf nejen edituje, ale i ukládá do databáze, načítá, maže atd.




-  Editace grafu – přidání vrcholu, mazání vrcholu, přidání hrany, mazání hrany
-  Uložení aktuálního stavu grafu do databáze. Jsou uloženy vrcholy, jejich pozice a označení, hrany a jejich ohodnocení
-  Vytvoření kopie aktuálního grafu do databáze s novým jménem.
-  Vytvoření nového grafu



Obr. 24 Náповěda

4 Ukázky

4.1 Přehled ukázek

V sekci je možné editovat komentované ukázky algoritmů. Editovanou ukázkou lze vybírat z přehledu všech ukázek. Ty jsou řazeny po jednotlivých grafech, k nimž jsou ukázky vztaženy.

Ukázky 02 – kostra	
1. Bipartita	 
2. Kruskalův algoritmus	 
3. Jarníkův algoritmus	 
4. Borůvkův algoritmus	 

Ukázky 03 – kostra, prohledávání	
1. Borůvkův algoritmus	 
2. DFS	 
3. BFS	 
4. Kruskalův algoritmus	 
5. Jarníkův algoritmus	 



Ukázky 09 – most	
1. Most	 

Ukázky 10 – most	
1. Most	 

Obr. 25 Přehled ukázek

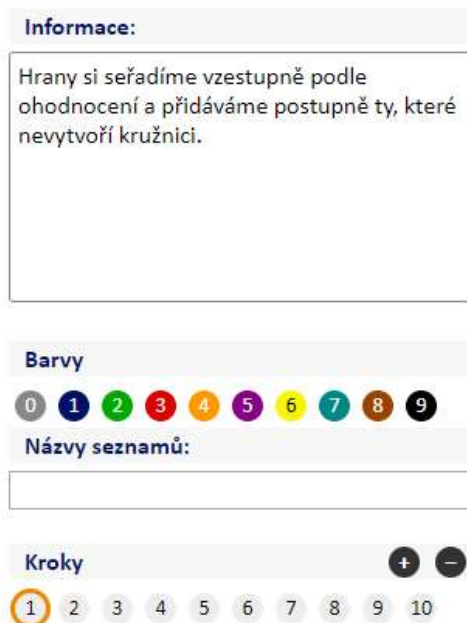
U každé ukázky je příznak vlajky, který značí, zda je ukázka zveřejněna nebo rozpracována (šedá vlaječka). Editovat lze pouze ukázky v rozpracovaném (neveřejněném) stavu. Ukázky lze přepnout kliknutím na symbol vlajky.

Ovládací prvky tabulky

-  změna stavu ukázky
-  smazání ukázky

4.2 Editace ukázky



Jak již bylo zmíněno, editovat (a mazat) lze jen ukázkou, která je v rozpracovaném stavu a má zašedlou ikonku dokončení. Editační stránka je rozdělena na dvě části. Vlevo je viditelný samotný graf a vpravo jsou ovládací prvky.



Obr. 26 Ovládání ukázky

Vpravo nahoře jsou pak ovládací prvky pro rychlou navigaci do přehledu grafů a ukázek a také přepínání mezi rozpracovaným a dokončeným stavem ukázky.

Ovládací prvky stránky

-  úprava vypisovaných seznamů vrcholů a hran
-  přepnutí stavu ukázky

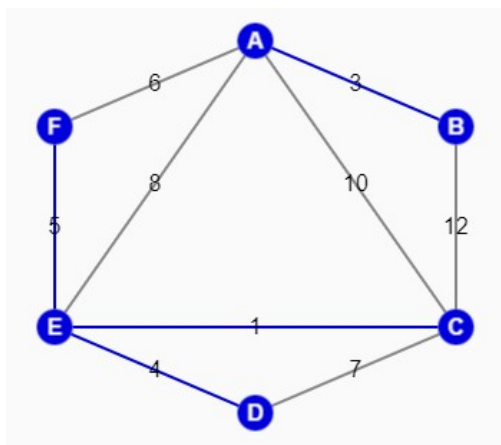
4.2.1 Nastavení editovaného kroku

Pro editaci ukázky nejdříve nastavíme krok, který chceme editovat. Uděláme to kliknutím na příslušné kolečko s číslem nebo šipkami vpravo a vlevo.



Obr. 27 Ovládání kroků

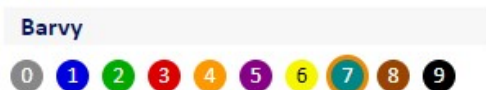
Graf se ihned přepne do podoby v tomto kroku.



Obr. 28 Zobrazení grafu v ukázce

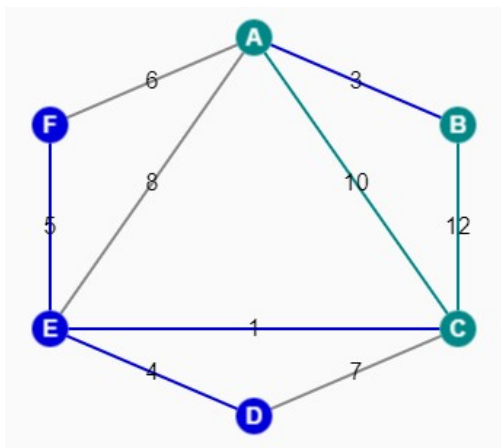
4.2.2 Nastavení barvy

Poté zvolíme barvu, kterou chceme v grafu měnit. Uděláme to opět kliknutím na kolečko s příslušnou barvou.



Obr. 29 Nastavení barvy u ukázky

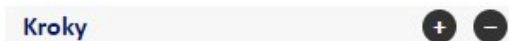
Po zvolení kroku a barvy můžeme editovat barevné nastavení grafu. Kliknutím na jednotlivý vrchol nebo hranu změníme jeho barevnou podobu.



Obr. 30 Více barev v grafu

4.2.3 Přidání nového kroku

Chceme-li v algoritmu přidat krok, uděláme to kliknutím na tlačítko „+“ (plus) v grafickém znázornění kroků.

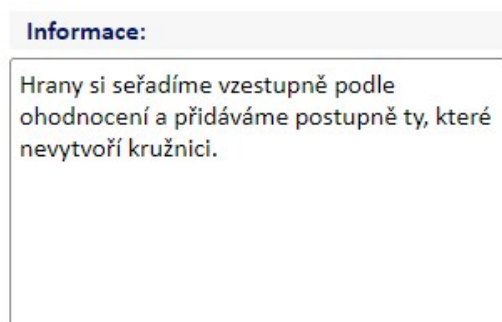


Obr. 31 Přidání a smazání kroku

Seznam kroků se rozšíří o další prvek, který se hned stane aktivním. Barevné nastavení grafu v nově přidaném kroku je zkopírováno z kroku dosud posledního. Poslední prvek odstraníme kliknutím na tlačítko „-“ (minus).

4.2.4 Vložení komentáře

Kromě nastavení barev lze u každého kroku vložit stručný popis (komentář). Ten lze zapsat do textového pole.²

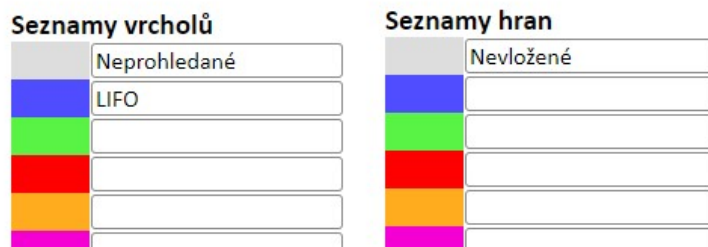


Obr. 32 Popisek kroku v ukázce

Nic z uvedeného není třeba ukládat. Barevné nastavení grafu i komentáře se automaticky ukládají do databáze.

4.3 Vypisované seznamy

Ke každé ukázce je možné nastavit a pojmenovat seznamy bodů a hran, které se pak v průběhu ukázky vypisují a písemně informují o stavu algoritmu.




Obr. 33 Seznamy vrcholů a hran

² Do textu nedoporučuji vkládat názvy barev (např. „Vrchol označíme zeleně“) vzhledem k tomu, že barvy lze editovat a text by po editaci barvy nemusel odpovídat realitě.

Ovládací prvky stránky

 uložení názvů seznamů a hran

 zpět na editaci ukázky

Vypsáním názvu seznamu se seznam bude v ukázce vypisovat, viz **Chyba! Nenalezen zdroj odkazů..** Např. všechny modré vrcholy se budou vypisovat v seznamu nazvaném LIFO.

5 Cvičení

5.1 Přehled cvičení

V sekci lze editovat cvičení sloužící k ověřování znalostí. Editované cvičení je možné vybírat z přehledu všech cvičení. Ta jsou řazena po jednotlivých grafech, k nimž jsou cvičení vztažena.

Cvičení (56)

Ukázky 03 – kostra, prohledávání	
1. DFS	 
2. BFS	 
3. Borůvkův algoritmus	 


Ukázky 05 – kostra, prohledávání	
1. BFS	 

Obr. 34 Výpis cvičení

U každého cvičení je příznak vlajky, který značí, zda je cvičení zveřejněno nebo rozpracované (šedá vlaječka). Cvičení lze editovat pouze v rozpracovaném stavu. Rozpracovaná cvičení jsou řazena jako první. Rozpracované cvičení lze také smazat pomocí ikonky koše.

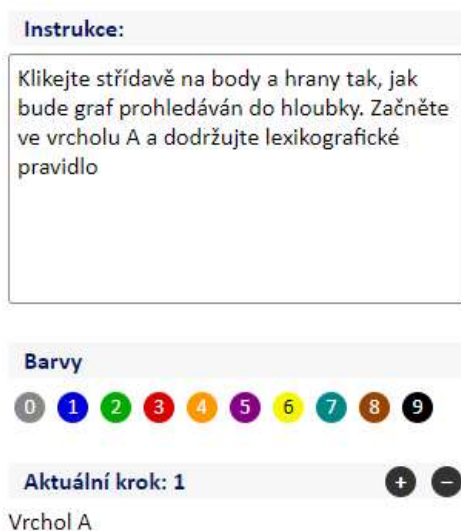
Ovládací prvky tabulky

 změna stavu cvičení

 smazání cvičení


5.2 Editace cvičení

Jak již bylo zmíněno, editovat (a mazat) lze jen cvičení, která je v rozpracovaném stavu a má zašedlou ikonku dokončení. Editační stránka je rozdělena na dvě části. Vlevo je viditelný samotný graf a vpravo jsou ovládací prvky.



Obr. 35 Ovládací prvky cvičení

Ovládací prvky stránky

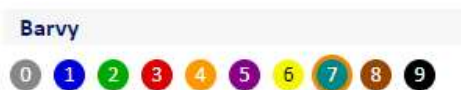
 přepnutí stavu cvičení

5.2.1 Editace instrukcí

V horní části lze editovat instrukce, které se zobrazí při spuštění cvičení. (Na rozdíl od ukázek je v textovém poli jediný text pro celé cvičení a nemění se při každém kroku.)

5.2.2 Nastavení barev

Barvu, která bude použita v příslušném kroku, lze nastavit kliknutím na příslušnou ikonku.

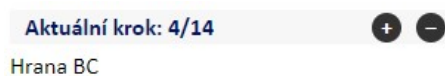


Obr. 36 Nastavení barev

Při cvičení lze v každém kroku provést pouze jednu akci (změnu barvy vrcholu nebo hrany). Proto se při kliknutí na element grafu tento element nejen obarví, ale je zrušena i původní akce (pokud taková byla).

5.2.3 Přidání nového kroku

Chceme-li ve cvičení přidat krok, uděláme to kliknutím na tlačítko „+“ v grafickém znázornění kroků.



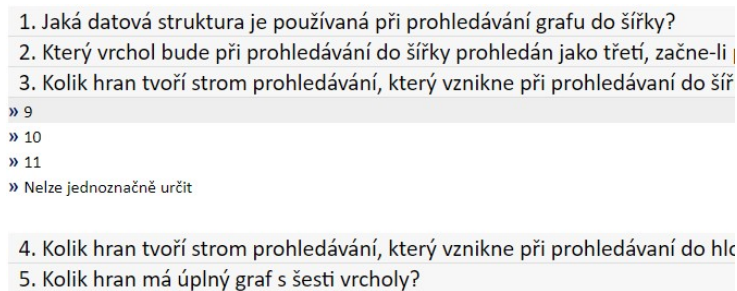
Obr. 37 Přidání nebo odstranění kroku

Počet kroků se rozšíří o další prvek. Poslední prvek odstraníme kliknutím na tlačítko „-“ (minus). Mezi jednotlivými kroky se pohybujeme pomocí šipek vpravo a vlevo. V elementu také vidíme, který krok aktuálně editujeme, dále celkový počet kroků a prvek grafu, který je editován. Nic z uvedeného není třeba ukládat, vše je ukládáno automaticky.

6 Testové otázky

6.1 Přehled testových otázek



V sekci *Test* je možné editovat testové otázky a odpovědi na ně. Editovanou testovou otázku je možné vybírat z přehledu všech otázek.





Obr. 38 Výpis testových otázek

Každou otázku je možné kliknutím rozbalit. Tím se zobrazí varianty odpovědí. Je také možné rozbalit všechny otázky současně kliknutím na znak šipky vpravo nahore.

Ovládací prvky tabulky

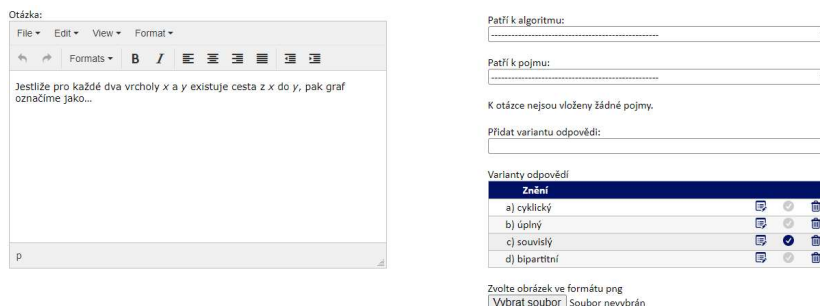
-  editace otázky
-  rozbalení/zabalení otázky

Ovládací prvky stránky

-  Vložení nové otázky
-  Rozbalení/zabalení všech otázek

6.2 Editace testové otázky

Editovat (a také přidávat a mazat) testovou otázku je možné v editační stránce po kliknutí na znak editace u zvolené otázky.








Obr. 39 Editace otázky

Editační stránka je rozdělena na dvě části. Editovat lze:

- znění otázky (včetně formátování),
- přiřazení k algoritmu, výběr ze seznamu – tato nepovinná hodnota zajistí vložení do testování konkrétního algoritmu,
- přiřazení k pojmu ve slovníku – nepovinná položka, která zajistí testování každého pojmu,
- obrázek – lze vkládat obrázek typu png, který je přiřazen k otázce.









Ovládací prvky stránky

-  vložení nové otázky
-  uložení změn provedených ve formuláři
-  přehled otázek
-  přepínání zveřejnění otázky
-  smazání otázky

6.3 Vkládání variant odpovědí



Na stejné stránce je možné vkládat varianty odpovědí k dané otázce. Všechny vložené varianty jsou vypsány a lze u nich označit správnou odpověď kliknutím na ikonku. Správná odpověď je u každé otázky právě jedna. Varianty odpovědí je také

možné smazat kliknutím na ikonku koše. Kliknutím na text odpovědi lze tuto odpověď editovat.

Varianty odpovědí	
Znění	
a) isomorfní	 
b) úplný	 
c) bipartitní	 
d) orientovaný	 

Obr. 40 Výpis variant odpovědí

Ovládací prvky tabulky

-  označení odpovědi za správnou
-  smazání odpovědi

6.4 Editace odpovědi v testu

Znění odpovědi na testovou otázku je možné upravit ve formuláři.



Editace odpovědi v testu

Jaká datová struktura je používána při prohledávání grafu do šířky?

Fronta LIFO

Obr. 41 Editace odpovědi na testovou otázku

Ovládací prvky stránky

-  uložení odpovědi
-  Zpět na editaci otázky

6.5 Vygenerování testu

Aplikace umožňuje vygenerovat test ve formátu PDF z uložených testových a otevřených otázek.

Ve formuláři je možná vyplnit:

- název testu,
- jméno,
- datum,
- počet testových (uzavřených) otázek,
- počet otevřených otázek

Název: <input type="text"/> Název testu	Jméno: <input type="text"/> Jméno
Datum: <input type="text"/> dd.mm.rrrr <input type="button" value="📅"/>	Počet uzavřených otázek: 1 <input type="button" value="▼"/>
<input type="button" value="Vygenerovat"/>	Počet otevřených otázek: 1 <input type="button" value="▼"/>

Obr. 42 Formulář pro vygenerování testu

Po odeslání formuláře bude vygenerován soubor s příslušným testem ve formátu PDF.

7 Otevřené testové otázky



7.1 Přehled otevřených testových otázek

V sekci otevřených testových otázek jsou tyto otázky uloženy za účelem generování testu ve formátu PDF. Jejich přehled je umístěn také v sekci *Test*.


1.	Rozhodněte o tom, zda je graf na obrázku eulerovský, a odůvodněte. V
2.	Rozhodněte o tom, zda je graf na obrázku eulerovský, a odůvodněte. V
3.	Vysvětlete pojem kostra grafu a minimální kostra grafu.
4.	Vysvětlete pojem kružnice grafu.
5.	Vysvětlete pojem souvislý graf.
6.	Vysvětlete pojem klikovost grafu.
7.	Vysvětlete pojem klika grafu.

Obr. 43 Otevřené testové otázky

Ovládací prvky tabulky

-  změna otevřené otázky
-  smazání otevřené otázky

Ovládací prvky stránky

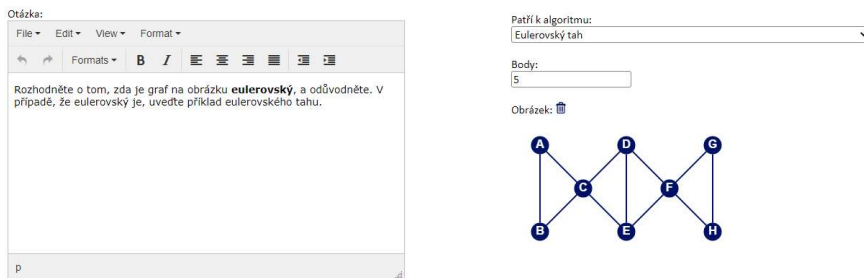
-  vložení nové otevřené otázky

7.2 Editace otevřených testových otázek

Kliknutím na zvolenou otázku je možné ji editovat. Lze změnit:

- znění otázky,
- přiřazení k algoritmu,

- počet bodů v testu,
- obrázek ve formátu png



Obr. 44 Editace otevřené testové otázky

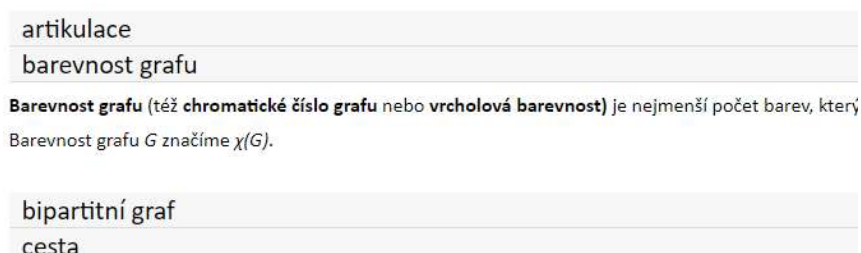
Ovládací prvky stránky

- vložení nové otevřené otázky,
- uložení změn provedených ve formuláři,
- přehled otevřených otázek,
- přepínání zveřejnění otevřené otázky,
- smazání otevřené otázky,

8 Slovník

8.1 Přehled pojmů

V sekci *Slovník* je možné editovat pojmy ve slovníku a jejich vysvětlení. Editovaný pojem je možné vybírat z přehledu všech pojmů.



Obr. 45 Výpis pojmů ve slovníku

Každý pojem je možné kliknutím rozbalit. Tím se zobrazí vysvětlení pojmu. Je také možné rozbalit všechny pojmy současně kliknutím na znak šipky vpravo nahoře.

Ovládací prvky tabulky

- Editace pojmu

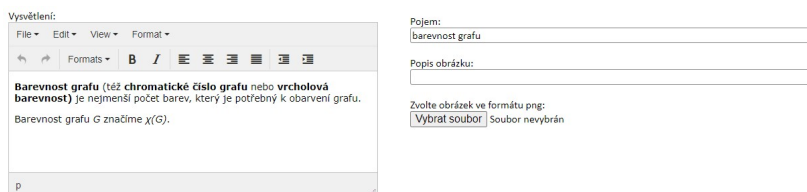
- ▼ Rozbalení/zabalení pojmu

Ovládací prvky stránky

- ⊕ Vložení nového pojmu
- ▼ Rozbalení/zabalení všech pojmů

8.2 Editace pojmu

Editovat (a také přidávat a mazat) pojmy ve slovníku je možné v editační stránce po kliknutí na znak editace u zvoleného pojmu. Editací stránka je rozdělena na dvě části.



Obr. 46 Editace pojmu ve slovníku

Editovat lze:

- vysvětlení pojmu (včetně formátování pomocí nástroje tinyMCE), jde o textovou položku.
- pojem – řetězec o maximální délce 100 znaků.
- obrázek – lze vkládat obrázek typu png, který je přiřazen k pojmu.
- popis obrázku – řetězec o maximální délce 100 znaků.

Prostřednictvím ovládacích prvků vpravo nahoře lze otázku uložit, smazat a přepínat mezi rozpracovaným dokončeným stavem. Mazání otázky je možné, i když obsahuje varianty odpovědí, ty jsou ale v tom případě smazány také. Další ovládací prvky umožňují rychlý přechod na přehled všech otázek a na stránku pro vložení otázky.

Ovládací prvky stránky

- ⊕ vložení nového pojmu
- 📄 uložení změn provedených ve formuláři
- ☰ přehled pojmů
- 🔍 přepínání zveřejnění pojmů
- 🗑️ smazání pojmu

9 Barvy

9.1 Přehled barev

V administrátorské sekci lze editovat jednotlivé barvy, které pak ovlivní vizuální podobu veškerých grafů. Editaci barev je proto nutné provádět obezřetně. Po otevření sekce barev je vidět jejich přehled včetně vizuální podoby.



Obr. 47 Výpis barev

9.2 Editace barvy

Barvy nelze přidávat ani mazat, je možné je jen editovat. To lze provést kliknutím na konkrétní barvu.

Forma editace barvy s následujícími prvky:

- Název barvy: textové pole obsahující 'Oranžová'
- Pozadí: barevný obdélník s oranžovou barvou
- Písmo: textové pole obsahující 'ABCD'
- Ukázka: barevný obdélník s oranžovou barvou a bílým textem 'ABCD'



Obr. 48 Editace barev

Lze měnit následující položky:

- název barvy, řetězec o maximální délce 30 znaků,
- kód barvy pozadí zvolením na paletě barev,
- kód barvy písma zvolením na paletě barev

Význam položek je zřejmý. Při editaci je třeba dbát na dobrý kontrast barvy a písma. Zvolené nastavení uložíme kliknutím na ikonku diskety v pravém horním rohu. Barev je celkem deset (z toho devět nastavitelných). Editovat nelze základní šedou barvu, která má v grafech označení „0“ (nula). Je nastavená pevně, její RGB hodnota je #888888 a hodnota barvy textu #ffffff (bílá). Nevhodnou změnou této barvy by mohlo dojít k celkovému znehodnocení vizualizace grafů.

Ovládací prvky stránky


-  uložení barvy
-  zpět na přehled barev

10 Uživatelé


10.1 Přehled uživatelů

V sekci *Uživatelé* je možné editovat jednotlivé uživatele, popřípadě je přidávat a mazat. Požadovaného uživatele je možné vybírat z přehledu všech uživatelů. Vzhled přehledu je analogický jako u předchozích přehledů položek.

Ovládací prvky tabulky

-  smazání uživatele

Ovládací prvky stránky

-  vložení nového uživatele

10.2 Editace uživatele

Editovat (a také přidávat a mazat) uživatele je možné v editační stránce.

E-mail:

Jméno:

Příjmení:

Heslo (min. 4 znaky, vyplňte jen při změně):





Heslo znovu pro kontrolu:

Obr. 49 Editace uživatele

Lze editovat:

- e-mail (slouží k přihlášení)
- jméno
- příjmení
- heslo
- heslo pro kontrolu

Ovládací prvky stránky

-  vložení nového uživatele
-  uložení změn provedených ve formuláři
-  přehled uživatelů
-  smazání uživatele

Příloha 2 – uživatelský manuál pro studenta

1 Úvod

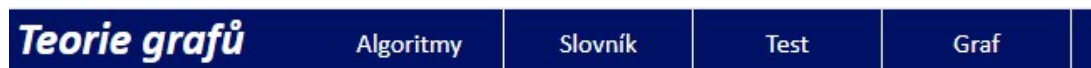
Webová aplikace teoriegrafu.cz je výuková aplikace, primárně určená pro studenty Fakulty informatiky a managementu Univerzity Hradec Králové. Tato uživatelská příručka obsahuje popis ovládání pro studenty.

1.1 Jednotlivé sekce aplikace

Studentská část aplikace teoriegrafu.cz obsahuje následující tematicky oddělené sekce, jejichž ovládání bude nadále popsáno.

- Algoritmy
- Slovník
- Test
- Graf

Jednotlivé sekce jsou také v menu v horní liště.



Obr. 1 Horní lišta s menu

1.2 Úvodní stránka

Na titulní stránce je přehled algoritmů z oblasti teorie grafů, jejichž přiblížení aplikace nabízí. U každého je stručný popis. Každý sektor slouží také jako odkaz, pomocí nějž se lze na stránku algoritmu dostat. Jde zároveň o úvodní stránku sekce *Algoritmy*.

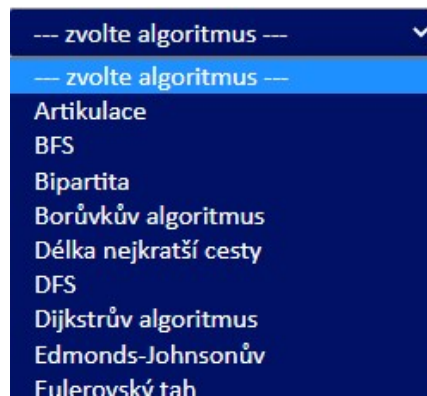


Obr. 2 Titulní stránka aplikace

2 Algoritmy

2.1 Přehled algoritmů

Jak již bylo uvedeno, je titulní stránka aplikace i úvodní stránkou sekce *Algoritmy*. Na ní je ale navíc rozbalovací menu pro rychlé přepínání mezi jednotlivými algoritmy bez nutnosti návratu na úvodní stránku. Toto rozbalovací menu zůstává viditelné na všech stránkách sekce.



Obr. 3 Nabídka algoritmů

2.2 Menu algoritmu

Pro každý algoritmus je v aplikaci nabídka nástrojů, která umožňuje se s ním podrobně seznámit. V této nabídce je:

- obecné,
- ukázka,
- video,
- cvičení,
- vyřešit,
- test



Obr. 4 Menu pro algoritmus

Následně budou jednotlivé položky z nabídky probrány.

2.3 Obecné

V části *Obecné* jsou textové informace o algoritmu a stručné vysvětlení jeho principů. Také jsou tu odkazy na další zdroje.

BFS – popis algoritmu**Anglický název**

Breadth-first Search

K čemu algoritmus slouží?

Prohledání vrcholů grafu, popř. nalezení konkrétní hodnoty vrcholu.

Časová složitost» $O(|V| + |E|)$ – každá hrana i vrchol jsou procházeny pouze jednou**Vstup**» souvislý graf $G(V,E)$ **Iniciaizace**

» libovolný vrchol vložený do fronty FIFO

Obr. 5 Textový popis algoritmu

2.4 Ukázka

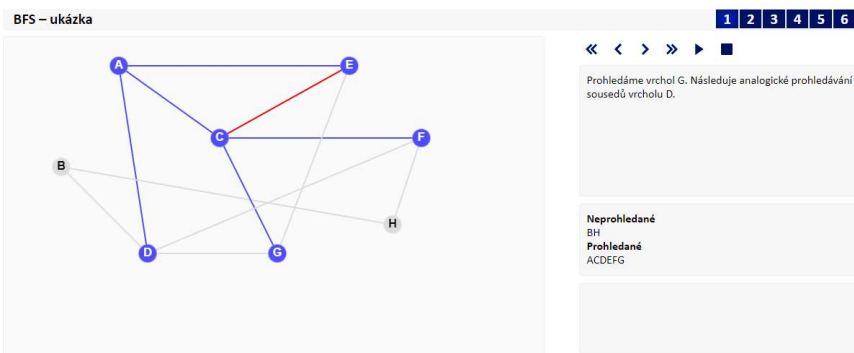
Pro algoritmy byly vytvořeny *Ukázky* jejich činnosti. Ty jsou v nabídce vpravo nahoře na úrovni názvu algoritmu a jejich počet se může pro každý algoritmus lišit.

BFS – ukázka

1 2 3 4 5 6

Obr. 6 Lišta s nabídkou ukázek vpravo

Po volbě ukázky se v levé části stránky vykreslí graf a v pravé části ovládací prvky a informační sektory.



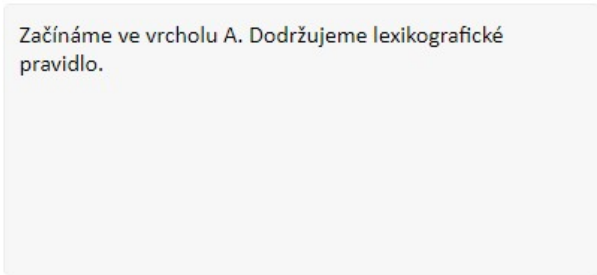
Obr. 7 Ukázka průchodu algoritmu

Ovládací prvky ukázky algoritmu

- ◀◀ skok na začátek algoritmu
- ◀ posun o jeden krok zpět (lze udělat klávesou šipky vlevo)
- ▶ posun o jeden krok vpřed (lze udělat klávesou šipky vpravo)
- ▶▶ Skok na konec algoritmu

- ▶ spuštění běhu algoritmu
- || pauza běhu algoritmu
- ukončení běhu algoritmu

Pomocí ovládacích prvků lze ovládat průběh algoritmu, opakovaně ho spouštět, zastavovat, krokovat a podobně. Ke každému kroku je uveden relevantní komentář v horním informačním sektoru.



Začínáme ve vrcholu A. Dodržujeme lexikografické pravidlo.


Obr. 8 Informačním sektor s komentářem

Další dva informační sektory jsou určeny k výpisu relevantních vrcholů a hran. Je tak možné sledovat změny, které nastanou v průběhu chodu algoritmu.



Neprohledané
BCDEFGH
Prohledané
A

Obr. 9 Informační sektor s přehledem vrcholů



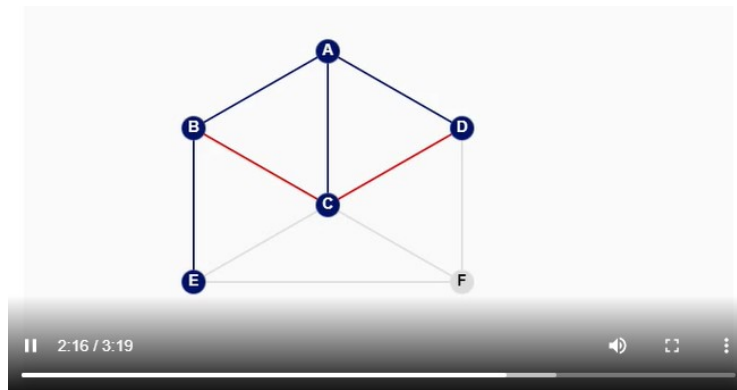
Stromové hrany
AD, AC, AE, BD, CF, CG
Nestromové hrany
CE, DF

Obr. 10 Informační sektor s přehledem hran

Díky ukázkám lze podrobně a opakovaně sledovat průběh nabízených algoritmů.

2.5 Video

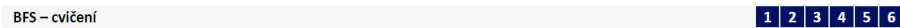
Video je další formou, jak pochopit princip algoritmu. Na nabízených videoukázkách je kromě samotného průběhu algoritmu i podrobný komentář, který vše vysvětluje. Video je možné ovládat běžně známými ovládacími prvky.



Obr. 11 Videoukázka algoritmu

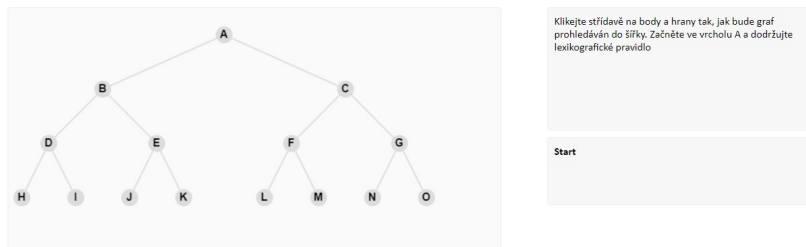
2.6 Cvičení

Svoje znalosti je možné si interaktivní formou vyzkoušet v sekci *Cvičení*. Podobně jako v ukázkách je jejich nabídka na úrovni názvu algoritmu napravo.



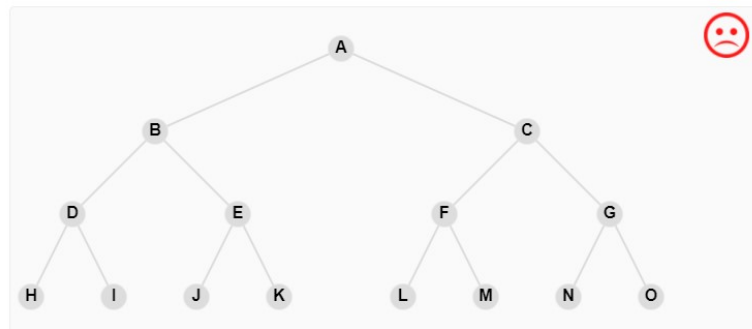
Obr. 12 Lišta s nabídkou cvičení vpravo

Po výběru konkrétního cvičení je v levé části vykreslen graf a v pravé části jsou uvedeny instrukce.

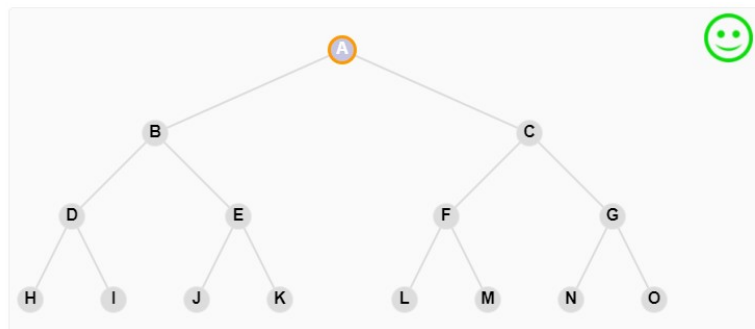


Obr. 13 Cvičení algoritmu

Úkolem je klikat podle instrukcí na elementy grafu v duchu průběhu algoritmu. Aplikace adekvátně reaguje buď zeleným smilečkem nebo červeným šklebíkem podle toho, jestli byla akce správná nebo došlo k chybě. To je doprovázeno akustickým signálem.



Obr. 14 Reakce na chybnou akci



Obr. 15 Reakce na správnou akci

Při správné akci (a tedy postupu v algoritmu) je navíc v informačním sektoru zobrazeno, k jaké akci došlo.

1. krok

Přidán vrchol A

Obr. 16 Popis správné akce při cvičení

Při správném průchodu celým algoritmem je úspěšné vyřešení cvičení oznámeno akustickým signálem a také výpisem slova „Hotovo“.

29. krok

Přidán vrchol O

Hotovo

Obr. 17 Úspěšné zakončení cvičení

2.7 Vyřešit

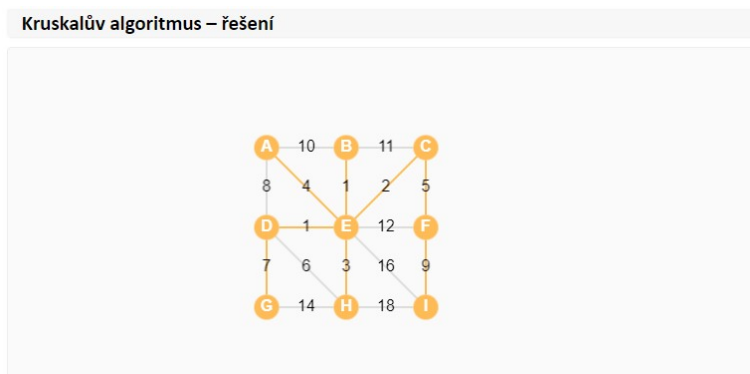
V sekci nazvané *Vyřešit* je možné spustit algoritmus grafu, který zadá uživatel. To je možné udělat v sekci *Graf*, viz dále. Pokud graf nebyl zadán, je zobrazen odkaz na sekci *Graf*, kde je to možné udělat.

BFS – řešení

Graf pro vyřešení můžete [zadat zde](#) nebo ho načíst ze souboru.

Obr. 18 Odkaz na stránku se zadáním grafu

Pokud je graf zadán, je možné si na něm příslušný algoritmus vyzkoušet. Algoritmus je možná spouštět, zastavovat i krokovat.



Obr. 19 Kruskalův algoritmus na vlastním grafu

V pravém informačním sektoru se postupně vypisují informace tak, jak algoritmus probíhá.

1. Přidaná hrana DE (váha: 1)
2. Přidaná hrana BE (váha: 1)
3. Přidaná hrana CE (váha: 2)
4. Přidaná hrana HE (váha: 3)
5. Přidaná hrana AE (váha: 4)
6. Přidaná hrana CF (váha: 5)
7. Přidaná hrana DG (váha: 7)
8. Přidaná hrana FI (váha: 9)

Obr. 20 Vypsání údajů v informačním sektoru

2.8 Test

Pokud existují testové otázky vztahující se k příslušnému algoritmu, lze je spustit jako test. Každá otázka má různý počet možných odpovědí, ale vždy jen jedna je správná. Úspěšnost odpovědí je v průběhu testu názorně vykreslována a vyjádřena v procentech.

4. Výsledkem prohledávání grafu do šířky je...

- a) Kostra grafu
- b) Fronta FIFO
- c) Fronta LIFO
- d) Nic z uvedeného

» Vyhodnocení »

75 %

Obr. 21 Testová otázka vztahující se ke konkrétnímu algoritmu

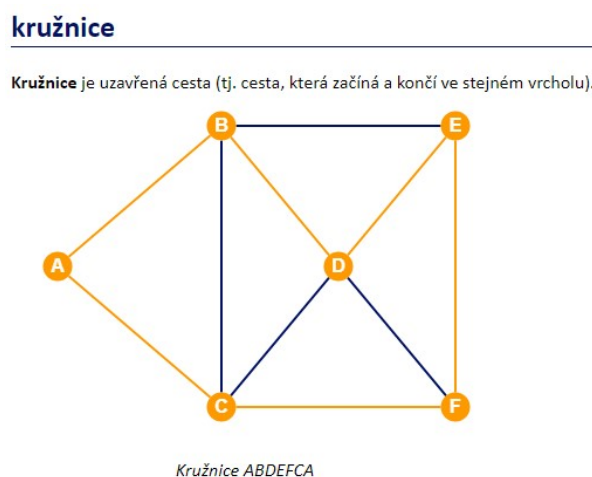
3 Slovník

V sekci *Slovník* jsou uvedeny termíny z teorie grafů a grafových algoritmů. Má hlavně informační charakter.

- | | |
|--|--|
| » artikulace | » komponenta souvislosti |
| » barevnost grafu | » koncový vrchol |
| » bipartitní graf | » kořenový strom |
| » cesta | » kostra grafu |
| » doplňěk (komplement) grafu | » kružnice |
| » eulerovský graf | » les |

Obr. 22 Přehled pojmů ve slovníku

Každý pojem je stručně vysvětlen, a umožňuje-li to charakter pojmu, vyobrazen na obrázku.



Obr. 23 Vysvětlení pojmu "kružnice"

Pokud jsou k pojmu přiřazeny otázky, lze pomocí testu znalost pojmu ověřit. Odkaz na test je vpravo, na úrovni nadpisu.

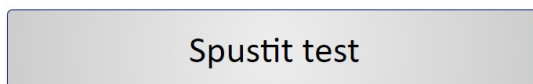
kružnice

TEST DEFINICE

Obr. 24 Odkaz na test

4 Test

Sekce *Test* je určena pro ověřování znalostí grafových algoritmů i pojmů. Princip jejího fungování je stejný jako u otázek k algoritmu nebo pojmu, viz kapitola 2.8. K zodpovězení jsou ale připraveny všechny otázky napříč celým tématem.



Obr. 25 Spuštění testu

Otázky mají různý počet možností, správná je ale jen jedna. Pořadí otázek je náhodné a uživatel je průběžně informován o své úspěšnosti.

7. Stupeň vrcholu, který je artikulací je...

- a) maximálně 1
- b) minimálně 1
- c) maximálně 2
- d) minimálně 2

» Další »



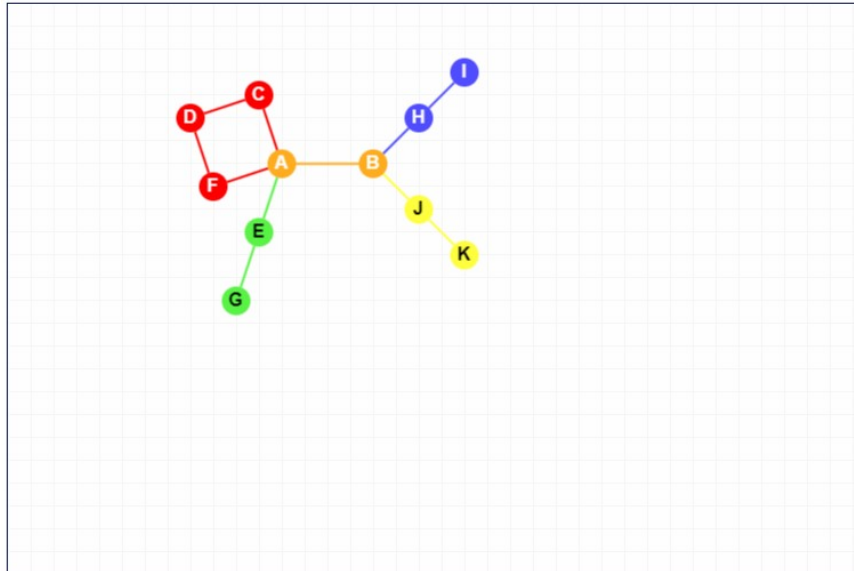
Obr. 26 Průběh testu

Po skončení testu je vypsáno vyhodnocení – počet i procento správných odpovědí.

5 Graf

V sekci *Graf* je možné vložit a upravovat vlastní graf, hlavně za účelem zkoušení algoritmů. Je možné graf také ukládat do souboru a zpětně ho ze souboru načíst.

Stránka sekce *Graf* je rozdělena na dvě části. V levé je vizualizace, vpravo ovládací prvky. Pro lepší umístění jednotlivých vrcholů je graf podložen mřížkou.



Obr. 27 Vizualizace grafu při editaci

Nastavení grafu

✖
📱
🖨
⬆

Žádná akce
Přidat vrcholy
Odebrat vrcholy
Přidat hrany
Odebrat hrany

0 1 2 3 4 5 6 7 8 9
🗑️ 📁 ⬇️

Vlastnosti grafu

- » orientovaný
- » ohodnocený
- » popis vrcholů

Ohodnocení hran

AB DE BE AD CF EF
CB

Popis vrcholů

A B C D E F





Vlastnosti

- » Vrcholů: 6
- » Hran: 7
- » Souvislost: ano
- » Komponent: 1

Obr. 28 Ovládací prvky grafu

5.1 Možnosti úprav grafu

Úpravy a manipulace s grafem jsou možné v několika oblastech, které charakterizují následující ovládací prvky.

-  základní úpravy – vkládání, mazání a přesuny vrcholů a hran, nastavení barev, editace označení
-  matice sousednosti
-  přímé odkazy na jednotlivé algoritmy
-  načtení grafu ze souboru

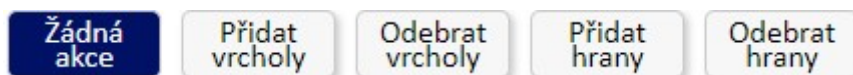
5.2 Základní úpravy

V této části lze editovat základní vlastnosti grafu.

- přidávat vrcholy,
- odebírat vrcholy,
- přidávat hrany
- odebírat hrany,
- přemísťovat vrcholy,
- přepínat orientaci grafu,
- přepínat ohodnocení hran a popis vrcholů,
- měnit ohodnocení hran a vrcholů popis vrcholů.

5.3 Manipulace s elementy grafu

Veškerou manipulaci s grafem lze provádět přepnutím do příslušného módu:



Obr. 29 Nastavení manipulace s vrcholy a hranami

5.3.1 Přidávání vrcholů

Vrchol lze přidat kliknutím na libovolné místo v grafu. Při tvorbě grafu je vhodné nejdříve přidat všechny vrcholy a následně k nim doplnit hrany.

5.3.2 Odebrání vrcholů

Vrchol je možné smazat kliknutím na něj. Kromě vrcholu jsou odebrány i všechny jeho incidenční hrany.

5.3.3 Přidání hrany

Hranu lze přidat postupným kliknutím na její počáteční a koncový vrchol. I pokud graf není orientovaný, je každá hrana uložena jako orientovaná (jen není orientace zobrazena).

5.3.4 Odebrání hrany

Kliknutím na příslušnou hranu ji lze odebrat.

5.3.5 Přemístění vrcholů

Vrcholy lze kdykoliv libovolně po grafu přemístit pomocí myši při nastavení módu „Žádná akce“. Spolu s vrcholem jsou přesunuty i incidenční hrany. Dalším následným kliknutím na vrchol je tento zarovnán do mřížky.

5.3.6 Přepínání orientace grafu

Vzhledem k tomu, že jsou hrany ukládány jako orientované, je možné snadným přepnutím checkboxu změnit orientaci grafu.

» orientovaný

Obr. 30 Nastavení orientace grafu

5.3.7 Přepínání hodnocení grafu

Graf je vždy ukládán jako ohodnocený, hodnocení hran ale mohou být skryta. Zda je graf ohodnocený lze snadno přepnout pomocí checkboxu:

» ohodnocený

Obr. 31 Nastavení ohodnocení grafu

5.3.8 Přepínání popisu vrcholů

Pomocí checkboxu lze také snadno přepínat, zda je viditelný popis vrcholů.

» popis vrcholů

Obr. 32 Nastavení popisu vrcholů

5.3.9 Změna ohodnocení hran

Pokud je nastaveno, že je graf ohodnocený, je možné jednotlivým hranám přiřadit celočíselné ohodnocení.

Ohodnocení hran							
AB	1	BC	5	CD	13	EF	15
FG	7	GH	10	IJ	6	JK	12
KL	3	AE	2	EI	8	BF	20
FJ	16	CG	14	GK	11	DH	9
HL	4						

Obr. 33 Vložení hodnocení hran

5.3.10 Změna popisu vrcholů

Pokud je nastaveno, že je jsou viditelné popisy vrcholů, je možné tyto popisy měnit jednoduchým přepsáním na textovou nebo číselnou hodnotu.

Popis vrcholů

A	B	C	D	E	F	G	H	I
---	---	---	---	---	---	---	---	---

Obr. 34 Změna popisu vrcholů grafu

5.3.11 Informace o vlastnostech grafu

Kromě toho jsou zde zobrazeny informace o grafu, konkrétně počet vrcholů, počet hran, souvislost grafu a počet komponent.




Vlastnosti

- » Vrcholů: 12
- » Hran: 17
- » Souvislost: ano
- » Komponent: 1

Obr. 35 Vlastnosti grafu

Všechny nastavené vlastnosti je nutné uložit kliknutím na ikonu diskety v pravé horní části okna.

V této části jsou také ovládací prvky pro manipulaci s grafem:

-  smazání grafu
-  uložení grafu
-  stažení grafu do souboru (musí být uložen)

5.4 Matice susednosti

Ve druhé části je viditelná matice susednosti grafu, která je generovaná z aktuálně uložené podoby grafu.

	A	B	C	D	E
A	0	5	0	3	8
B	0	0	1	2	0
C	0	0	0	7	0
D	0	0	0	0	0
E	0	4	0	5	0

Obr. 36 Matice susednosti

5.5 Odkazy

V třetí části je přehled odkazů na příslušné sekce algoritmů, kde lze na graf aplikovat některý z algoritmů.

Algoritmy

[Artikulace](#)
[BFS](#)
[Bipartita](#)
[Borůvkův algoritmus](#)
[Délka nejkratší cesty](#)
[DFS](#)
[Dijkstrův algoritmus](#)
[Edmonds-Johnsonův](#)
[Eulerovský tah](#)
[Jarníkův algoritmus](#)
[Kruskalův algoritmus](#)
[Most](#)
[Počet komponent](#)
[Souvislost grafu](#)
[Tarryho algoritmus](#)
[Trémauxův algoritmus](#)
[Vrchol na kružnici](#)
[Vrcholy v komponentě](#)

Obr. 37 Odkazy na algoritmy

5.6 Načtení grafu ze souboru

Čtvrtá část umožňuje načíst graf ze souboru, kam byl předtím uložen.

Graf:

Soubor nevybrán

Obr. 38 Načtení grafu ze souboru

Zadání diplomové práce

Autor: Bc. Aleš Krupička
Studium: I14290
Studijní program: N1802 Aplikovaná informatika
Studijní obor: Aplikovaná informatika

Název diplomové práce: Podpora výuky předmětu Diskrétní metody a optimalizace.

Název diplomové práce Support of the course Discrete methods and optimization.
AJ:

Cíl, metody, literatura, předpoklady:

Cílem práce je vytvořit ucelenou webovou aplikaci pro podporu výuky předmětu Diskrétní metody a optimalizace. Aplikace bude názornou formou demonstrovat probírané algoritmy na základě zadání uživatele a také ilustrativní příklady. 1. Nastudování potřebné části teorie grafů 2. Teoretické zpracování grafových algoritmů 3. Navržení aplikace 4. Seznámení se softwarem, v kterém bude nástroj vytvářen 5. Vytvoření aplikace 6. Popis a ovládaní aplikace 7. Sestavení databáze příkladu vhodných pro výuku a procvičování 8. Testování aplikace

[1] Matoušek J., Nešetřil J. (2007): Kapitoly z diskretní matematiky, Karolinum, Praha [2] Šišma P. (1997): Teorie grafů 1736-1963, Prometheus, Praha [3] Opava Z. (1989): Matematika kolem nás, Albatros, Praha [4] Anderson I. (2001): A First Course in Discrete Mathematics, Springer, London [5] Robertson N., Sanders D.P., Seymour P.D., Thomas R. (1997): The Four Color Theorem, Journal of Combinatorial Theory Ser. B 70, 2-44 [6] Jirovský L. (2008): Vybrané problémy z teorie grafů ve výuce na střední škole, bakalářská práce, MFF UK, Praha [7] Šedivý J. (1966): O konečných grafech a jejich využití, časopis Matematika ve škole, ročník XVI [8] Biggs N. (2002): Discrete mathematics, Oxford University Press [9] Turzík D., Pavlíková P. (2007): Diskrétní matematika, VŠCHT, Praha

Garantující pracoviště: Katedra informatiky a kvantitativních metod,
Fakulta informatiky a managementu

Vedoucí práce: RNDr. Andrea Ševčíková, Ph.D.

Oponent: prof. RNDr. PhDr. Antonín Slabý, CSc.

Datum zadání závěrečné práce: 14.1.2015