

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Využití grafových algoritmů v herních aplikacích a hlavolamech
Ondřej Pokorný

Bakalářská práce
2023

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ondřej Pokorný**
Osobní číslo: **I20145**
Studijní program: **B0688A140009 Informační technologie**
Téma práce: **Využití grafových algoritmů v herních aplikacích a hlavolamech**
Zadávající katedra: **Katedra informačních technologií**

Zásady pro vypracování

Mnoho her a hlavolamů je založeno na některých typech úloh z teorie grafů. Jako příklad můžeme uvést např. kreslení jedním tahem (Eulerův graf), přesun vrcholů tak, aby se hrany spojující vrcholy nekřížily (rovinné grafy), prohledávání bludiště nebo barvení politických map (problém 4 barev). Teorii grafů zde lze využít buď při tvorbě úlohy či kontrole řešení.

Cílem této práce bude výběr vhodných her a hlavolamů, kde lze teorii grafů využít.

Cílem praktické části bude implementace teoretické části a tvorba herní aplikace založené na grafových algoritmech.

Rozsah pracovní zprávy: **30 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

BALAKRISHNAN, V. *Schaum's outline of theory and problems of graph theory*. New York: McGraw-Hill, c1997, viii, 293 p. ISBN 00-700-5489-4.

TÖPFER, Pavel. *Algoritmy a programovací techniky*. 1. vyd. Praha: Prometheus, 1995, 299 s. ISBN 80-858-4983-6.

MATOUŠEK, Jiří. *Kapitoly z diskrétní matematiky*. Vyd. 1. Praha: Karolinum, 2002, 381 s. ISBN 80-246-0084-6.

Vedoucí bakalářské práce: **RNDr. Josef Rak, Ph.D.**
Katedra matematiky a fyziky

Datum zadání bakalářské práce: **16. prosince 2022**
Termín odevzdání bakalářské práce: **12. května 2023**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 28. února 2023

Prohlašuji:

Práci s názvem Využití grafových algoritmů v herních aplikacích a hlavolamech jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 9. 5. 2023

Ondřej Pokorný

PODĚKOVÁNÍ

Chtěl bych poděkovat všem, kteří mi v průběhu mé bakalářské práce pomohli a podpořili mě. Zejména bych rád poděkoval svému vedoucímu práce RNDr. Josefu Rakovi Ph.D. za jeho cenné rady, trpělivost a podporu. Dále bych rád poděkoval své rodině a přátelům za neustálou motivaci a povzbuzení, které mi pomohly dokončit tuto práci.

ANOTACE

Mnoho her a hlavolamů je založeno na některých typech úloh z teorie grafů. Jako příklad můžeme uvést např. kreslení jedním tahem (Eulerův graf), přesun vrcholů tak, aby se hrany spojující vrcholy nekřížili (rovinné grafy), prohledávání bludiště nebo barvení politických map (problém 4 barev). Teorii grafů zde lze využít buď při tvorbě úlohy či kontrole řešení. Cílem této práce bude výběr vhodných her a hlavolamů, kde lze teorii grafů využít. Cílem praktické části bude implementace teoretické části a tvorba herní aplikace založené na grafových algoritmech.

KLÍČOVÁ SLOVA

Teorie grafů, graf, vrchol, hrana, Eulerův tah, Hamiltonovská kružnice, Rovinný graf, Barvení grafu

TITLE

The use of graph algorithms in game applications and puzzles

ANNOTATION

Many games and puzzles are based on some types of graph theory problems. Examples include drawing in one stroke (Euler graph), moving vertices so that the edges connecting the vertices do not cross (planar graphs), searching a maze, or coloring political maps (the 4-color problem). Graph theory can be used here to either create the problem or check the solution. The aim of this paper will be to select appropriate games and puzzles where graph theory can be used. The practical part will aim at implementing the theoretical part and creating a game application based on graph algorithms.

KEYWORDS

Graph theory, graph, vertex, edge, Euler move, Hamiltonian circle, Plane graph, Graph coloring

OBSAH

Seznam obrázků.....	10
Seznam zkratk.....	13
Úvod.....	14
1 Úvod do teorie grafů.....	15
1.1 Graf.....	15
1.2 Historie.....	16
1.2.1 Vznik teorie grafů.....	16
1.2.2 19. století.....	16
1.2.3 Počátek 20. století.....	17
1.2.4 Druhá polovina 20. století.....	17
1.3 Základní pojmy Teorie grafů.....	17
1.3.1 Definice: Sled, cesta, tah, dráha, kružnice.....	17
1.3.2 Podgraf.....	18
1.3.3 Isomorfismus.....	19
1.3.4 Orientovaný graf.....	20
1.3.5 Neorientovaný graf.....	20
1.3.6 Stupeň vrcholu.....	21
1.3.7 Kompletní graf.....	22
1.3.8 Souvislý graf.....	22
1.3.9 Multigraf.....	24
1.3.10 Rovinný graf.....	24
2 Vybrané hry a Využití algoritmy.....	26
2.1 Jednotažky – eulerovské grafy.....	26
2.1.1 Problém sedmi mostů města Královce – historický problém.....	27
2.1.2 Eulerovský tah.....	28
2.1.3 Fleuryho algoritmus.....	29
2.1.4 Ukázka Fleuryho algoritmu.....	29
2.1.5 Aplikace Eulerova tahu.....	33
2.2 Hamiltonová kružnice.....	34
2.2.1 Hamiltonovský graf.....	34

2.2.2	Postačující podmínky existence hamiltonovské kružnice	34
2.2.3	Nutné podmínky existence hamiltonovské kružnice	35
2.2.4	Algoritmus pro hledání hamiltonovské kružnice.....	36
2.2.5	Příklad backtracking algoritmu.....	36
2.2.6	Icosian game	38
2.2.7	Obchodní cestující	39
2.2.8	Jezdcova procházka	39
2.3	Rovinné zobrazení grafu.....	40
2.4	Barvení rovinného grafu – problém čtyř barev.....	42
2.4.1	Problém v teorii grafů	43
2.4.2	Historie problému čtyř barev	43
2.4.3	Barvení grafu	44
2.4.4	Algoritmus pro barvení grafu	45
2.4.5	Ukázka sekvenčního barvení	46
2.4.6	Další aplikace barvení grafu	50
2.5	Algoritmy k otestování souvislosti grafu.....	51
2.5.1	Průchod grafu do hloubky.....	51
2.5.2	Příklad průchodu do hloubky.....	52
2.5.3	Průchod grafu do šířky.....	54
2.5.4	Příklad průchodu do šířky.....	54
3	Popis herní aplikace.....	57
3.1	Prerekvizity pro spuštění aplikace	57
3.2	Popis aplikace	57
3.3	Eulerovský tah	58
3.4	Hamiltonovská kružnice.....	60
3.5	Hledání rovinného zobrazení	61
3.6	Barvení grafu	62
4	Technická dokumentace.....	64
4.1	Použité technologie.....	64
4.1.1	Java	64
4.1.2	JavaFX	64

4.1.3	Gluon Scene Builder	64
4.1.4	NetBeans IDE 8.2	64
4.2	Rozvržení aplikace.....	64
4.3	Popis vybraných částí kódu	66
4.3.1	Generování náhodného grafu.....	66
4.3.2	Prohlídka do šířky	66
4.3.3	Hledání Eulerova tahu	67
4.3.4	Hledání hamiltonovské kružnice	67
4.3.5	Barvení grafu	68
Závěr		69
Použitá literatura		70

SEZNAM OBRÁZKŮ

Obrázek 1: Graf G a jeho indukovaný podgraf H.....	18
Obrázek 2: Graf G a jeho obecný podgraf H.....	19
Obrázek 3: Grafy G a H (inspirováno zdrojem [8]).....	19
Obrázek 4: Orientovaný graf.....	20
Obrázek 5: Neorientovaný graf.....	21
Obrázek 6: Graf se stupni vrcholů 3, 3, 2, 3, 4, 1.....	21
Obrázek 7: Úplný graf obsahující tři vrcholy.....	22
Obrázek 8: Úplný graf obsahující čtyři vrcholy.....	22
Obrázek 9: Souvislý graf (inspirováno zdrojem [3]).....	23
Obrázek 10: Nesouvislý graf (inspirováno zdrojem [3]).....	23
Obrázek 11: Multigraf a prostý graf.....	24
Obrázek 12: Rovinný graf.....	25
Obrázek 13: Nerovinný graf.....	25
Obrázek 14: Rozložení města Královec [14].....	27
Obrázek 15: Graf sedmi mostů města Královce.....	28
Obrázek 16: Graf pro ukázkou Fleuryho algoritmu.....	30
Obrázek 17: Určení stupně vrcholů.....	30
Obrázek 18: Zvolení počátečního vrcholu.....	31
Obrázek 19: Vybrání hrany.....	31
Obrázek 20: Vybírání hran, co nejsou mosty.....	32
Obrázek 21: Dvě komponenty souvislosti.....	32
Obrázek 22: Možné řešení Eulerova tahu.....	33
Obrázek 23: Petersenův graf (inspirováno zdrojem [18]).....	35
Obrázek 24: Graf G pro ukázkou backtracking algoritmu.....	36
Obrázek 25: Ukázkou backtracking algoritmu č. 1.....	37
Obrázek 26: Ukázkou backtracking algoritmu č. 2.....	37
Obrázek 27: Hamiltonovská kružnice v grafu G.....	38
Obrázek 28: The Icosian Game[20].....	38
Obrázek 29: Hamiltonovská kružnice ve dvanáctistěnu (inspirováno zdrojem [18]).....	39
Obrázek 30: Příklad uzavřené jezdcovi procházky [21].....	40
Obrázek 31: Nerovinné grafy K_5 a $K_{3,3}$ (inspirováno zdrojem [19]).....	41
Obrázek 32: Graf G a jeho rozdělení vzniklé půlením hran.....	42

Obrázek 33: Příklady homeomorfních grafů K_5 a $K_{3,3}$ (inspirováno zdrojem [19]).....	42
Obrázek 34: Politická mapa Evropy [3]	43
Obrázek 35: Příklad vrcholového barvení grafu.....	44
Obrázek 36: Mapa střední Evropy[3]	46
Obrázek 37: Popsání států[3].....	46
Obrázek 38: Převedení mapy do grafu	47
Obrázek 39: Určení stupně vrcholů	47
Obrázek 40: Obarvení vrcholu N.....	48
Obrázek 41: Obarvení vrcholu R.....	48
Obrázek 42: Obarvení vrcholu Č.....	49
Obrázek 43: Výsledné obarvení grafu	49
Obrázek 44: Obarvená mapa (zdroj původního obrázku: [3]).....	50
Obrázek 45: Příklad procházení grafu do hloubky (inspirováno zdrojem [25]).....	52
Obrázek 46: První krok prohlídky do hloubky	52
Obrázek 47: Druhý krok průchodu do hloubky	53
Obrázek 48: Třetí krok prohlídky do hloubky	53
Obrázek 49: Čtvrtý krok prohlídky do hloubky.....	53
Obrázek 50: První krok prohlídky do šířky	55
Obrázek 51: Druhý krok prohlídky do šířky.....	55
Obrázek 52: Třetí krok průchodu do šířky.....	56
Obrázek 53: Hlavní menu aplikace.....	57
Obrázek 54: Okno Eulerův tah	58
Obrázek 55: Spojování bodů.....	59
Obrázek 56: Správné řešení	59
Obrázek 57: Špatné řešení	60
Obrázek 58: Okno hamiltonovská kružnice.....	60
Obrázek 59: Okno rovnání grafu	61
Obrázek 60: Okno řešení rovnání grafu.....	62
Obrázek 61: Okno barvení grafu	62
Obrázek 62: Balíčky	65
Obrázek 63: Generování náhodného grafu kód	66
Obrázek 64: Prohlídka grafu do šířky kód.....	66
Obrázek 65: Hledání Eulerova tahu kód.....	67
Obrázek 66: Hledání hamiltonovské kružnice kód.....	67

SEZNAM ZKRATEK

BFS	Breadth First Search
DFS	Depth First Search
JRE	Java Runtime Environment
GPS	Global Positioning System
FF	First Fit
LDO	Large Degree Ordering

ÚVOD

Teorie grafů je důležitou disciplínou matematiky, která má velmi mnoho užití nejen v informatice, ale i v přírodních a sociálních vědách a mnoha dalších situacích v reálném světě. Dalším z využití teorie grafů je aplikace grafových algoritmů v různých hrách a hlavolamech.

Hlavním cílem této práce je prozkoumat, jaké algoritmy z teorie grafů lze při vytváření různých her a hlavolamů použít a jakým způsobem je můžeme v tvorbě her a hlavolamů využít. Dalším cílem této práce je tyto algoritmy využít k implementaci jednoduché herní aplikace, která bude pro hráče jak zábavou, tak i výzvou.

Tato práce je zaměřená na čtyři hlavní problémy teorie grafů, Eulerovský tah, Hamiltonovská kružnice, hledání rovinného zobrazení grafu a barvení rovinného grafu.

Hlavním důvodem psaní této práce bylo vytvořit herní aplikaci, která využívá tyto čtyři algoritmy k vytvoření čtyř různých her. Jelikož je teorie grafů využívána ve spousta případech v reálném světě, je herní aplikace dobrým způsobem, jak obeznámit veřejnost s aspoň nějakým základem teorie grafů touto zábavnou formou. Herní aplikace je psána v jazyce Java. První implementovanou hrou je hledání Eulerovského tahu ve vygenerovaném grafu, kde hráč postupně prochází hrany grafu, dokud nesestrojí Eulerovský tah. Druhou implementovanou hrou je hledání Hamiltonovské kružnice ve vygenerovaném grafu, která funguje na stejném principu jako Eulerův tah. Třetí implementovanou hrou je hledání rovinného zobrazení grafu, kdy hráč musí narovnat graf tak, aby se hrany grafu nekřížily. Poslední hrou je barvení rovinného grafu čtyřmi barvami, kdy musí hráč barvit graf do té doby, než je obarven tak, aby dva sousední vrcholy neměly stejnou barvu.

Herní aplikace umožňuje hrát všechny tyto čtyři hry ve třech různých obtížnostech, tak aby si hráč mohl zvolit obtížnost podle jeho potřeby. Dále umožňuje herní aplikace zobrazit řešení dané hry, aby si hráč, který si neví rady mohl nechat poradit a také se zároveň něco přiučil.

V teoretické části je nejprve krátce popsána historie teorie grafů a následně vysvětleny některé základní pojmy teorie grafů. Dále jsou v teoretické části popsány všechny použité algoritmy. Každý z těchto algoritmů je názorně předveden na jednoduchém příkladu. Ke každému z použitých algoritmů je uvedeno několik příkladů, jak se dá daný algoritmus využít v praxi.

1 ÚVOD DO TEORIE GRAFŮ

Teorie grafů se svými mnohými užitími v přírodních i sociálních vědách a v teoretické informatice obzvláště, je důležitou součástí učebních osnov matematiky všech škol a univerzit po celém světě. [1]

Mnoho struktur včetně situací z reálného světa může být přesvědčivě reprezentováno na papíře prostřednictvím diagramu, který se skládá z množiny bodů (obvykle značené malými kruhy nebo tečkami) společně s čarami spojující některé, nebo všechny dvojice těchto bodů. Například body v diagramu mohou reprezentovat různá města v nějakém státě, a čára, která spojuje dva body, ale neprochází přes třetí bod, indikuje například, že mezi městy, reprezentovanými těmito dvěma body, existuje přímý letový spoj. V některých případech se může stát, že existuje přímý letový spoj z města A do města B, ale neexistuje přímý letový spoj z města B do města A. V těchto situacích se používá šipka na jednom konci spojnice, která značí, že je tato spojnice orientovaná. Matematická abstrakce takovýchto struktur obsahující body a spojnice těchto bodů nás vede ke konceptu grafů a diagramů. [1]

1.1 Graf

V diskrétní matematice je pojem grafu klíčovým pojmem. Graf je algebraická struktura, která umožňuje přehledně popisovat objekty a jejich vztahy. Graf si můžeme představit jako několik bodů (vrcholů) spojených čarami (hranami). Tento způsob zobrazení je intuitivní a jednoduchý, ale pro detailní analýzu a implementaci algoritmů nestačí. Proto je nutné definovat graf jako strukturu, která popisuje objekty a jejich vztahy. Pokud mezi dvěma objekty existuje vztah, zaznamenáme ho jako dvojici objektů (vrcholů), zatímco pokud neexistuje, nezahrneme tuto dvojici do množiny hran. Tento přístup umožňuje přesně popsat strukturu grafu bez ohledu na jeho zobrazení. [2]

Definice 1.1: *Graf G je uspořádaná dvojice (V, E) , kde V je nějaká neprázdná množina a E je množina dvoubodových podmnožin množiny V . Prvky množiny V se jmenují **vrcholy** grafu G a prvky množiny E **hrany** grafu G . [3]*

1.2 Historie

1.2.1 Vznik teorie grafů

Mezi matematickými disciplínami je teorie grafů jedna z těch relativně mladších v porovnání s jinými disciplínami, jako jsou aritmetika nebo geometrie, jejichž počátky sahají daleko před začátkem letopočtu. Teorie grafů začala být systematicky studována až v 18. století, kdy švýcarský a pruský matematik Leonhard Euler objasnil tzv. „Problém sedmi mostů města Královce“, který spočívalo v určení, zda je možné projít všechny sedm mostů města Královce právě jednou. Euler dokázal, že v tomto specifickém případě to nelze, avšak problém zobecnil na jakékoliv počty mostů, či míst, na které mosty vedou a ukázal, že projít každý most právě jednou, lze jen v případě, že ze všech míst vychází sudý počet mostů, anebo právě ze dvou míst vychází lichý počet mostů. Euler využil novou disciplínu, kterou nazval "geometrii pozic", ale kterou bychom dnes označili jako teorii grafů. Euler je proto považován za zakladatele teorie grafů a rok 1736 je považován za rok vzniku této disciplíny. Nicméně další významná práce v oblasti teorie grafů byla provedena až více než sto let poté. [2]

1.2.2 19. století

Trvalo více než sto let od toho, co Euler pracoval na problému sedmi mostů města Královce, než byla v teorii grafů vykonána další významná práce. Kolem roku 1850 se začaly v teorii grafů objevovat nové problémy v nečekaných oblastech včetně chemie. Pomalu rostoucí výskyt grafů v matematice podpořil další studium grafů. V roce 1878 byl britským matematikem Josephem Sylvesterem zaveden termín „graf“. Poprvé ho zmínil v časopise Nature, který se zabýval souvislostmi mezi matematikou a chemií. [4]

V roce 1847 německý fyzik G.R. Kirchhoff přišel s teorií koster grafů. Nejprve našel kostru v grafu, který reprezentoval elektrickou síť, a následně přidával různé hrany tak, aby uzavřeli různé obvody a pro každý tento obvod sestavil lineární rovnici. Díky tomu zformuloval zákony, které se používají pro výpočet napětí a proudu v jednotlivých větvích elektrického obvodu. O 10 let později britský matematik Arthur Cayley využil teorie grafů k počítání organických izomer C_nH_{2n+2} . Za vrcholy označil jednotlivé atomy a za hrany označil chemickou vazbu mezi atomy a následně ukázal, že molekuly alkanů jsou stromy. [5][19][8]

1.2.3 Počátek 20. století

Velká část základů moderní teorie grafů byla vytvořena na počátku 20. století. Grafy již přestaly být považovány za kuriozitu na okraji matematiky, ale staly se oborem matematiky. Toto vyplývá z faktu, že se začaly objevovat knihy na téma teorie grafů. První oficiální kniha byla vydána v roce 1936 maďarským matematikem Dénesem Königem. Byla publikována v němčině pod názvem „Theorie der endlichen und unendlichen Graphen.“ [4]

1.2.4 Druhá polovina 20. století

S vývojem počítačů v polovině dvacátého století, se výrazně posunula i teorie grafů. Studování grafů se nejen stalo jednodušším, ale grafy jako takové našly mnoho nových uplatnění ve výpočetní technice a v počítačových sítích. Na konci padesátých let se v oblasti grafů uskutečnili významné práce. Maďarští matematici Paul Erdős and Alfréd Rényi začali tak zvaný pravděpodobnostní přístup k teorii grafů. Na základě toho vznikl nový obor, který je znám jako teorie náhodných grafů. [5]

1.3 Základní pojmy Teorie grafů

1.3.1 Definice: Sled, cesta, tah, dráha, kružnice

Sled je posloupnost vrcholů a hran $\{v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n\}$ ve kterém každá hrana e_i vede z vrcholu v_{i-1} do vrcholu v_i ($e_i = [v_{i-1}, v_i]$). [7]

*Sled, ve kterém se neopakují hrany se nazývá **tah**. Tah, ve kterém se neopakují vrcholy se nazývá **cesta**. [9]*

*Uzavřená neorientovaná cesta se nazývá **kružnice**. Orientovaný i neorientovaný sled, ve kterém se neopakují vrcholy se nazývá **tah**. [7]*

*Orientovaný sled, ve kterém se neopakují hrany se nazývá **dráha**. Dráha, která začíná i končí ve stejném vrcholu se nazývá **cyklus**. [7]*

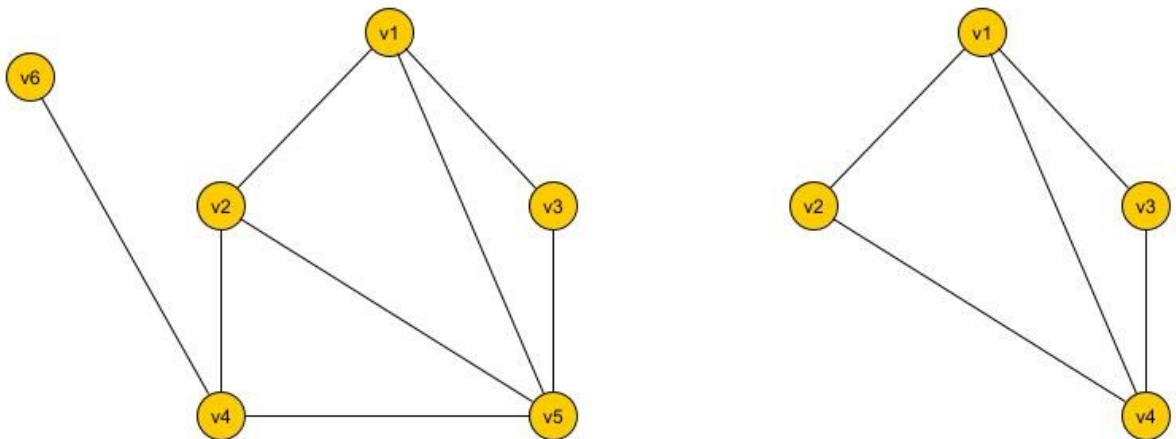
1.3.2 Podgraf

Při řešení různých problémů často uvažujeme případy, kdy z daného grafu vynecháním nějakých vrcholů (a hran se kterými jsou incidentní), nebo nějakých hran, případně obojího, dostaneme jeho část. Dostáváme se tak k termínu „podgraf“. [2]

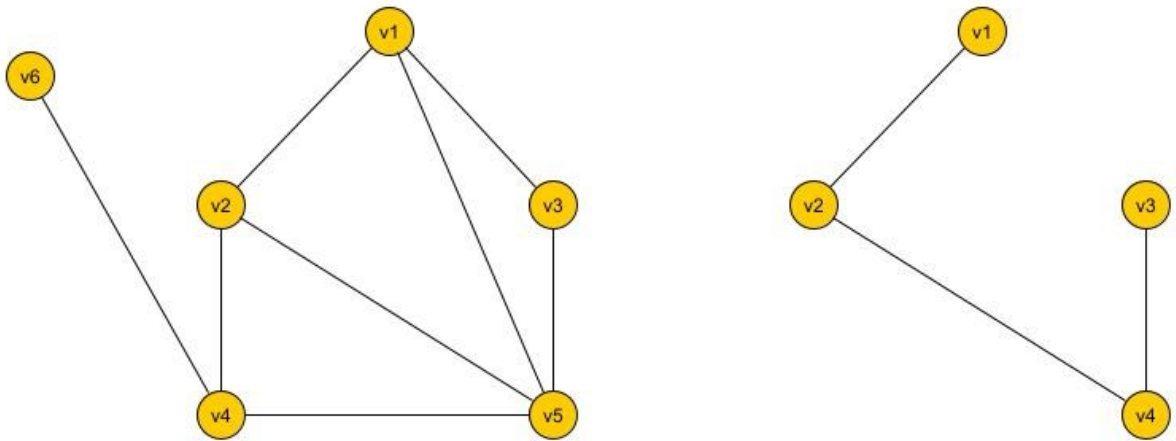
Definice 1.2: Graf H nazveme **podgrafem** grafu G , jestliže $V(H) \subseteq V(G)$ a $E(H) \subseteq E(G)$. Pišeme $H \subseteq G$. [2]

Při odebrání vrcholu je nutné odebrat také všechny hrany, které vedou do anebo z tohoto vrcholu. Pokud byly odebrány jen tyto hrany, jde o tak zvaný indukovaný podgraf. Pokud byly odebrány i jiné hrany, jde obecně o podgraf. [8]

Definice 1.3: Podgraf I grafu G nazveme **indukovaným podgrafem** grafu G , jestliže $E(I)$ obsahuje všechny hrany grafu G , které jsou incidentní s vrcholy z $V(I)$. (Vynecháme pouze hrany, které byli incidentní s vynechanými vrcholy.) [2]



Obrázek 1: Graf G a jeho indukovaný podgraf H

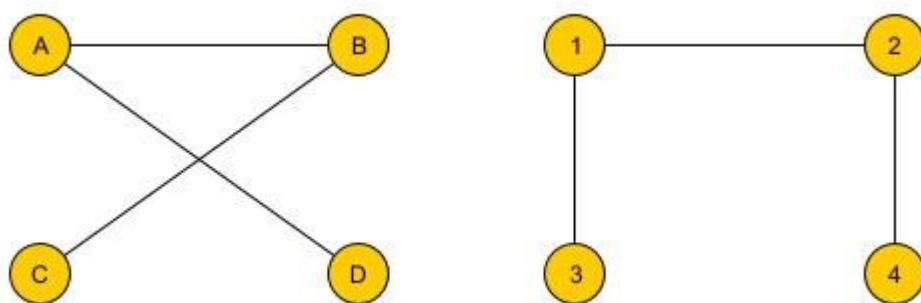


Obrázek 2: Graf G a jeho obecný podgraf H

1.3.3 Isomorfismus

Modelování reálných situací pomocí grafů je výhodné, protože se zaměřujeme na základní vlastnosti struktury a abstrahuje od konkrétního provedení. Nicméně stejnou situaci lze popsat různými grafy, například při jiném pojmenování vrcholů nebo jiném zakreslení grafu. Abychom mohli rozeznat, zda dva grafy mají stejnou strukturu, ale liší se pouze označením nebo nakreslením, používáme pojem isomorfismus. [2]

Definice 1.4: *Isomorfismus grafů G a H je bijektivní zobrazení $f: V(G) \rightarrow V(H)$, pro které platí, že každé dva vrcholy u, v v grafu G jsou sousední právě tehdy, když jsou sousední jejich obrazy $f(u), f(v)$ v grafu H . [2]*



Obrázek 3: Grafy G a H (inspirováno zdrojem [8])

Grafy G a H na obrázku nahoře jsou izomorfní, zobrazení $f: V(G) \rightarrow V(H)$ dané seznamem vrcholů a jejich obrazů zachovává sousednost. [2]

$$f(A) = 1$$

$$f(B) = 2$$

$$f(C) = 4$$

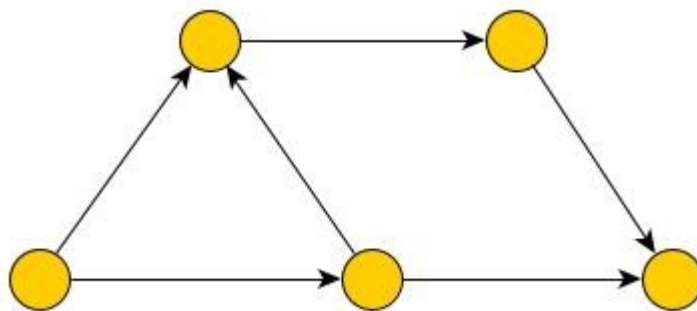
$$f(D) = 3$$

Vrcholy A a B stupně 2 se zobrazí na vrcholy 1 a 2 stupně 2. Vrcholy C a D stupně 1 se zobrazí na vrcholy 4 a 3 stupně 1. [2]

1.3.4 Orientovaný graf

Definice 1.5: *Orientovaný graf* je trojice $G = (V, E, \varepsilon)$ tvořená konečnou množinou prvků V , jejíž prvky nazýváme uzly, konečnou množinou E , jejíž prvky nazýváme orientovanými hranami a zobrazením $\varepsilon : E \rightarrow V^2$, které nazýváme vztahem **incidence**, a které přiřazuje každé hraně $e \in E$ uspořádanou dvojici uzlů. [6]

Orientované grafy mají především využití pro popis vztahů, které jsou asymetrické v jejich podstatě. Tyto vztahy mohou zahrnovat například nadřazenost jednoho prvku vůči druhému, návaznost jedné činnosti na druhou, podskupinu jedné skupiny v rámci druhé skupiny, vítězství jednoho týmu nad druhým a podobně. [7]



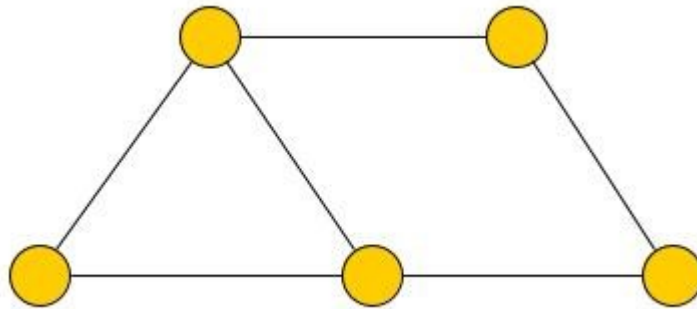
Obrázek 4: Orientovaný graf

1.3.5 Neorientovaný graf

Definice 1.6: *Neorientovaný graf* je trojice $G = (V, E, \varepsilon)$ tvořená konečnou množinou V , jejíž prvky nazýváme uzly, konečnou množinou E , jejíž prvky nazýváme neorientovanými hranami a zobrazením ε (vztah **incidence**), které přiřazuje každé hraně e jedno nebo dvouprvkovou množinu uzlů. [6]

Neorientované grafy jsou nejvhodnější pro popis symetrických vztahů, například sousedství mezi dvěma městy. Avšak v některých situacích není zcela jasné, zda se jedná o symetrický nebo asymetrický vztah, jako například v případě přímé cesty mezi dvěma místy. V takovém

případě je obvykle vhodné použít neorientovaný graf. Pokud však na této cestě existují jednosměrné úseky, jedná se o systém, kde je alespoň jedna část potřeba popsat orientovanou hranou, a proto je vhodné použít orientovaný graf pro celý systém. [7]

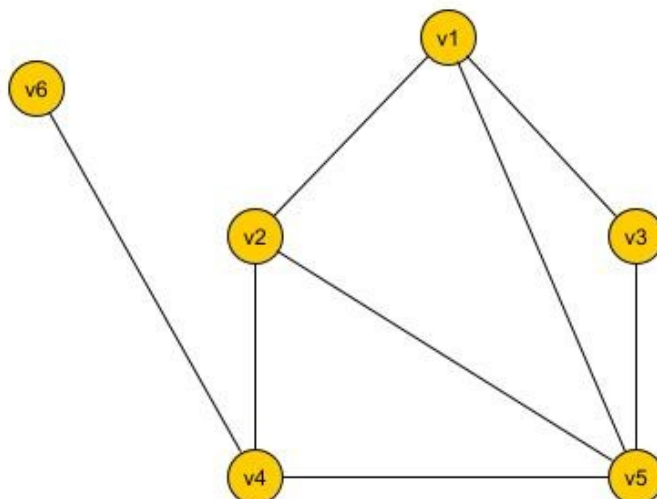


Obrázek 5: Neorientovaný graf

1.3.6 Stupeň vrcholu

Definice 1.7: *Stupeň vrcholu* v grafu G je definován jako počet hran, se kterými je vrchol incidentní. [2]

Určení stupně vrcholů v grafu na obrázku je velmi snadné. Stačí spočítat počet incidentních hran každého vrcholu. Stupně vrcholu je však možné určit i v případě, že je graf zadán množinami vrcholů V a hran E nebo je zadán jménem. [2]

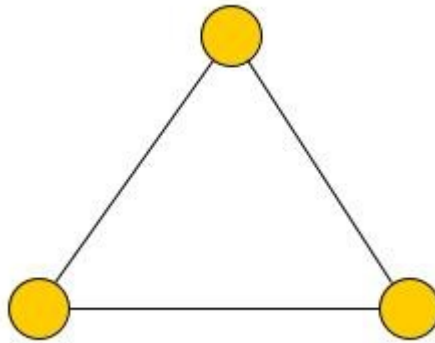


Obrázek 6: Graf se stupni vrcholů 3, 3, 2, 3, 4, 1

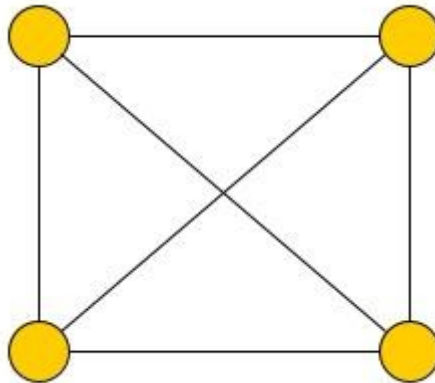
1.3.7 Kompletní graf

Kompletní graf je takový graf, který má všechny dvojice vrcholů propojené hranou. [8]

Definice 1.8: Graf na n vrcholech, kde $n \in \mathbb{N}$, který obsahuje všech $\binom{n}{2}$ hran se nazývá **úplný** nebo také **kompletní graf** a značí se K_n . [2]



Obrázek 7: Úplný graf obsahující tři vrcholy



Obrázek 8: Úplný graf obsahující čtyři vrcholy

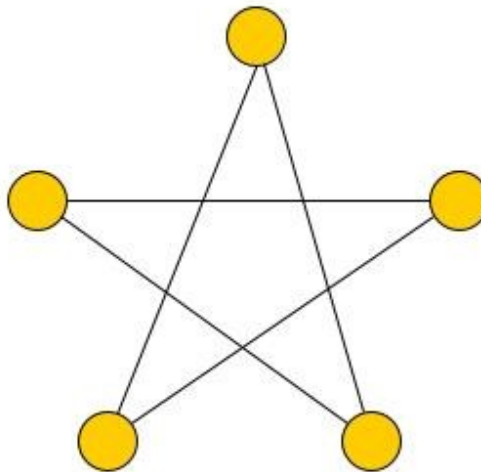
1.3.8 Souvislý graf

Definice 1.9: Řekneme, že vrchol v je dosažitelný z vrcholu u , jestliže v grafu existuje sled z vrcholu u do vrcholu v . Graf nazveme **souvislý**, jestliže pro každé dva vrcholy u, v je vrchol v dosažitelný z vrcholu u . V opačném případě je graf **nesouvislý**. [2]

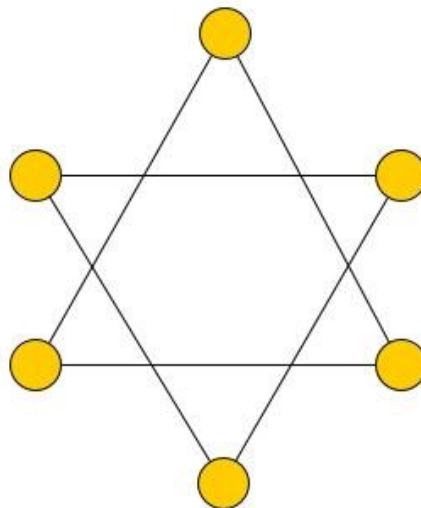
Definice 1.10: **Komponenta souvislosti** je každý v inkluzi maximální souvislý podgraf. Souvislé grafy jsou právě ty, co mají pouze jednu komponentu souvislosti. U orientovaných grafů se navíc rozlišují silně souvislé komponenty. [27]

Definice 1.11: Neorientovaný graf je souvislý, pokud je každá dvojice vrcholů spojena neorientovanou cestou. Největší možný souvislý podgraf grafu, je komponenta souvislosti grafu. Orientovaný graf je **silně souvislý**, pokud mezi každými dvěma vrcholy u a v existuje orientovaná cesta z u do v i z v do u , přičemž obě cesty mohou vést jinudy. [7]

Definice 1.12: Artikulace je takový vrchol, který když vyjmeme, zvýšíme počet komponent grafu minimálně o jednu. Most je taková hrana, kterou když vyjmeme, zvýšíme počet komponent grafu právě o jednu. [10]



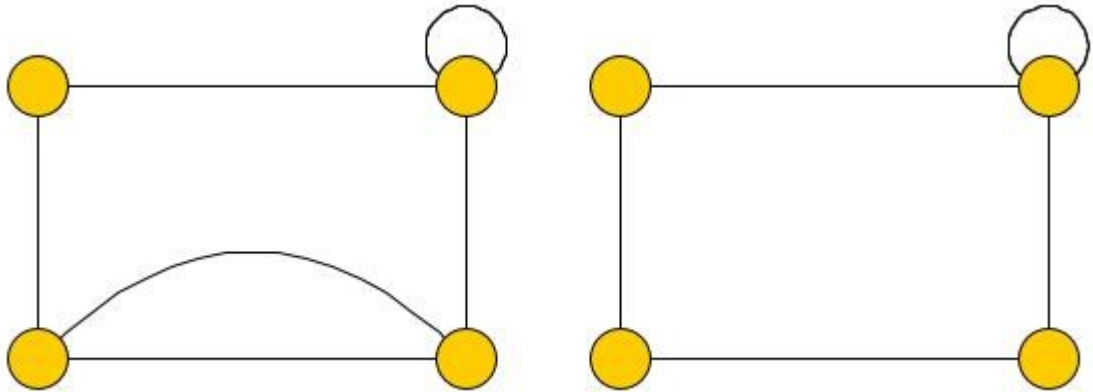
Obrázek 9: Souvislý graf (inspirováno zdrojem [3])



Obrázek 10: Nesouvislý graf (inspirováno zdrojem [3])

1.3.9 Multigraf

Definice 1.13: *Multigraf* je takový graf, který obsahuje mezi některou dvojicí vrcholů více hran v jednom směru. To znamená, že graf může obsahovat rovnoběžné hrany nebo smyčky (hrana která spojuje jeden stejný vrchol). Graf, který neobsahuje žádné smyčky ani násobné hrany se nazývá **jednoduchý graf**. Graf, který neobsahuje žádné násobné hrany se nazývá **prostý graf**. [6]

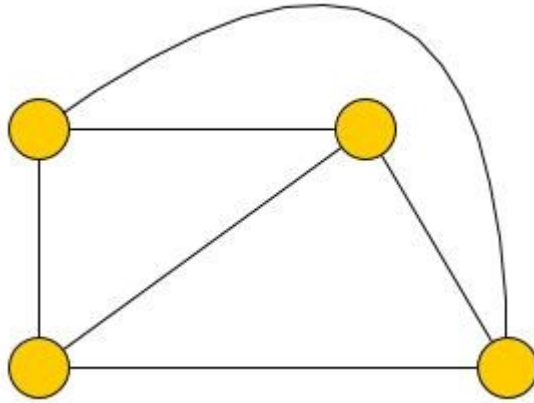


Obrázek 11: Multigraf a prostý graf

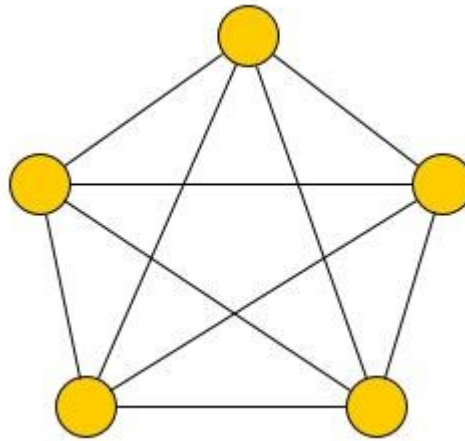
1.3.10 Rovinný graf

Definice 1.15: *Rovinný graf* je graf, který je možné nakreslit na rovnou plochu tak, aby se žádné jeho dvě hrany nekřížily. V tomto nakreslení jsou vrcholy v rovině, hrany mezi vrcholy nemají společný žádný bod, kromě koncových bodů. [11]

Většinu rovinných grafů lze nakreslit tak, aby se hrany křížily, stačí když existuje jedno rovinné zobrazení, aby byl graf rovinný. Velké množství grafů, které jsou z reálného světa jsou rovinné, například silnice bez nadjezdů, plošné spoje nebo mapy států. [11]



Obrázek 12: Rovinný graf



Obrázek 13: Nerovinný graf

2 VYBRANÉ HRY A VYUŽITÉ ALGORITMY

V informačních technologiích se grafy využívají k reprezentaci komunikačních sítí, organizaci dat nebo výpočtu toku. Například struktura odkazů webových stránek může být reprezentována pomocí orientovaného grafu, kde vrcholy představují webové stránky a orientované hrany představují odkazy z jedné stránky na druhou. Například ve fyzice a chemii mohou grafy představovat složité trojrozměrné struktury a molekuly, což usnadňuje jejich studium. V biologii a ochraně přírody se grafy používají k mapování oblastí, kde se vyskytují určité druhy, které jsou znázorněny vrcholem a jejich migrace nebo cesty, kterými se zvířata pohybují, reprezentovány hranami mezi oblastmi, což pomáhá při sledování jejich chování a šíření. Tyto informace mohou být důležité při určování způsobů rozmnožování a šíření druhů, nemocí, parazitů a jak mohou změny pohybu ovlivnit jiné druhy. Využití teorie grafů se neomezuje jen na tyto oblasti, ale může být aplikována v mnoha jiných disciplínách, kde může poskytnout užitečné nástroje pro analýzu a řešení složitých problémů. [12]

Dalším z možných využití teorie grafů může být tvorba jednoduchých her a hlavolamů, které jsou založeny na teorii grafů. Tato práce se věnuje především čtyřem hlavním hrám a algoritmům, které tyto hry využívají, a to jsou kreslení jedním tahem (Eulerův tah), hledání hamiltonovské kružnice, hledání rovinného zobrazení rovinného grafu, a barvení rovinného grafu čtyřmi barvami.

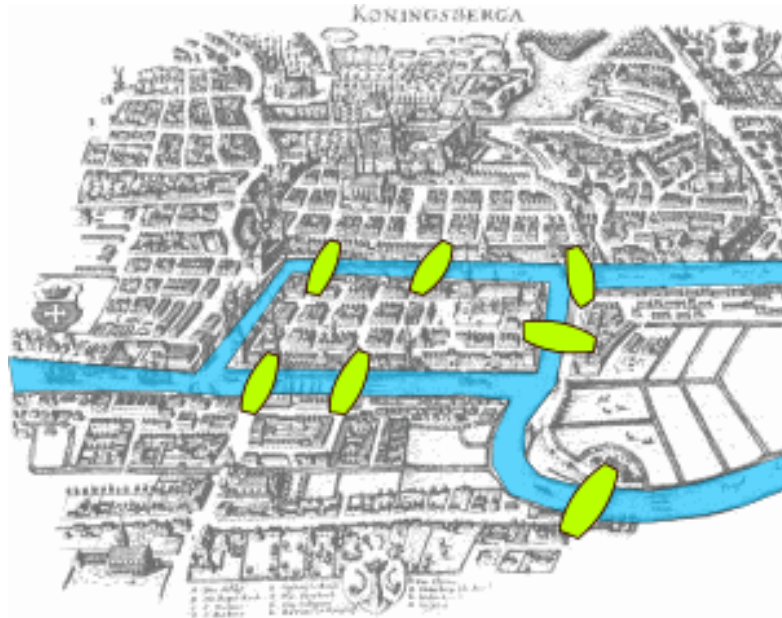
2.1 Jednotažky – eulerovské grafy

Jednou z nejstarších a základních hádanek týkající se teorie grafů je úloha, která spočívá v nakreslení daného obrázku (grafu) jedinou uzavřenou linií, aniž byste přitom zvedli tužku z papíru a aby se každá čára (hrana) v tahu objevila právě jednou a každý bod alespoň jednou. Matematicky řečeno, je třeba najít uzavřený sled, kde se každá hrana vyskytuje právě jednou a každý vrchol minimálně jednou. Takovýto sled nazýváme uzavřený eulerovský tah. Pokud graf obsahuje alespoň jeden takový tah, nazýváme ho eulerovským grafem. [3]

Eulerovský graf poprvé zavedl Leonhard Euler, když se snažil vyřešit velmi známý historický problém sedmi mostů města Královce. [2]

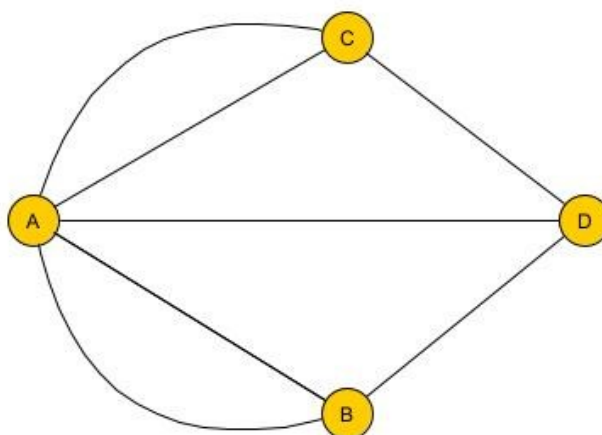
2.1.1 Problém sedmi mostů města Královce – historický problém

V 18. století vznikl známý problém v Pruském městě Královec, které se táhne na obou stranách řeky Pregola. V centru města je ostrov s názvem Kneiphof, ke kterému vede sedm mostů spojujících ho s ostatními částmi města. Rozložení města je vidět na obrázku dole. [13]



Obrázek 14: Rozložení města Královec [14]

Lidé města Královce se rozhodli si pro sebe vytvořit hru, úkol byl najít takovou cestu okolo města tak, aby po každém mostě přešli pouze jednou. I když se nikomu z obyvatel města nepodařilo takovou cestu najít, nebyli schopni to dokázat. To dokázal v roce 1735 Leonhard Euler, kdy prezentoval práci, která obsahovala řešení sedmi mostů města Královce, ale také obecné řešení pro jakýkoliv počet kusů pevniny a jakýkoliv počet mostů. Euler si pojmenoval mosty a, b, c, d, e, f a g. A pevniny označil A, B, C, D. Pro lepší představu si problém převedeme do grafu. [13]



Obrázek 15: Graf sedmi mostů města Královce

Euler dokázal, že neexistuje taková cesta grafem, aby se každá hrana opakovala jen jednou jinými slovy neexistuje uzavřený eulerovský tah. Dále Euler vysvětlil, že je jasné, že pokud by zde byly přesně dva kusy pevniny, s lichým počtem mostů vedoucích z nich, takovou cestu nalézt lze. Ale protože ze všech kusů pevniny v Královci vede lichý počet mostů, tak v tomto případě taková cesta neexistuje. Po tom, co teorie grafů zažila „boom“ po tom, co Euler vyřešil tento problém, se v roce 1875 lidé Královce rozhodli postavit nový most, mezi pevninami B a D, čímž se zvýšil počet mostů vedoucích z těchto kusů pevnin ze 3 na 4, tím pádem zůstaly pouze dva mosty s lichým počtem mostů a cestu bylo možné najít. Dnešní Královec má bohužel po škodách způsobených za 2. světové války mostů už jen 5. [13]

2.1.2 Eulerovský tah

Jak už bylo řečeno Eulerův tah je takový tah, kde je každá hrana navštívena právě jednou. Eulerův tah se běžně hledají v neorientovaných grafech. Eulerův tah v grafu existuje, pokud graf splňuje dvě podmínky. Graf musí být spojitý, a musí mít buď žádný nebo právě dva vrcholy, které mají lichý počet hran a ostatní vrcholy musí mít lichý počet hran. Pokud graf tyto podmínky splňuje, označujeme ho jako Eulerovský graf. Eulerovský tah rozdělujeme na uzavřený a otevřený Eulerovský tah. [15]

- Uzavřený Eulerovský tah: začíná i končí ve stejném vrcholu
- Otevřený Eulerovský tah: počáteční a koncový vrchol tahu se liší

Uzavřený Eulerovský tah v neorientovaném grafu existuje právě tehdy, když splňuje následující podmínky:

- Je spojitý
- Všechny jeho vrcholy jsou sudého stupně

Otevřený Eulerovský tah v grafu existuje právě tehdy když splňuje následující podmínky:

- Je spojitý
- Obsahuje právě dva vrcholy lichého stupně, všechny ostatní vrcholy jsou stupně sudého [15]

2.1.3 Fleuryho algoritmus

K nalezení uzavřeného Eulerovského tahu můžeme použít Fleuryho algoritmus [27].

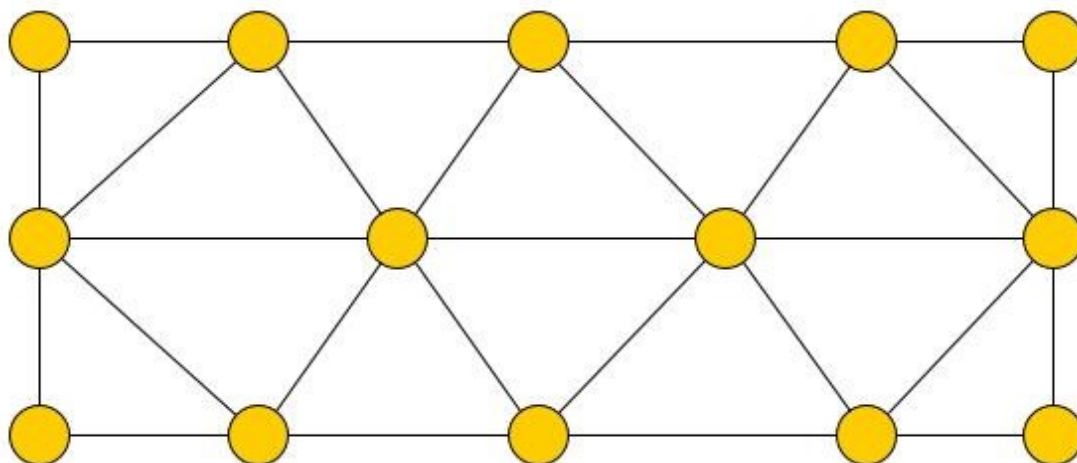
Fleuryho algoritmus:

- (1) Zvolíme libovolný vrchol u v grafu. Vybereme libovolnou hranu h , která je incidentní s vrcholem u , tuto hranu projdeme a označíme jako navštívenou.
- (2) Jsou-li označeny všechny hrany, algoritmus končí.
- (3) Zvolíme nenavštívenou hranu, která je incidentní s naposledy navštíveným vrcholem, touto hranou projdeme a označíme ji jako navštívenou. Hranu volíme tak, aby po označení této hrany nedošlo k rozpadu podgrafu, který je složený z dosud neoznačených hran a vrcholů, které jsou s těmito hranami incidentní na:
 - a. Dvě neprázdné komponenty souvislosti
 - b. Neprázdnou komponentu souvislosti a vrchol, ve kterém jsme eulerovský tah začali
- (4) Vrátime se zpět na krok (2). [27]

Fleuryho algoritmus lze použít i v případě, že graf obsahuje otevřený eulerovský tah, tedy obsahuje právě dva vrcholy lichého stupně. Tyto dva vrcholy lichého stupně propojíme pomocnou hranou. Touto pomocnou hranou projdeme při provádění Fleuryho algoritmu jako první. V moment, kdy dokončíme uzavřený eulerovský tah, odebereme z tohoto tahu hranu, kterou jsme na začátku přidali. Tímto způsobem dostaneme otevřený eulerovský tah. [27]

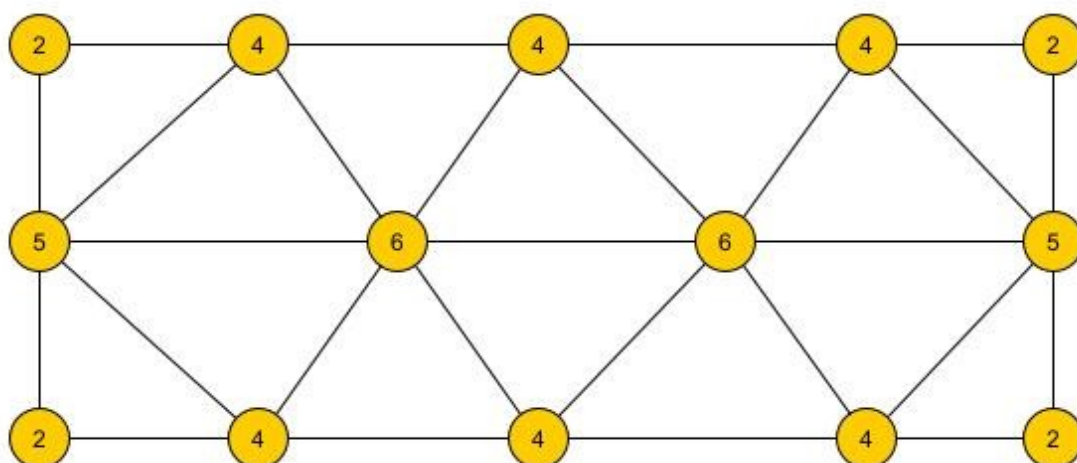
2.1.4 Ukázka Fleuryho algoritmu

Fleuryho algoritmus si ukážeme na následujícím grafu viz Obrázek 16.



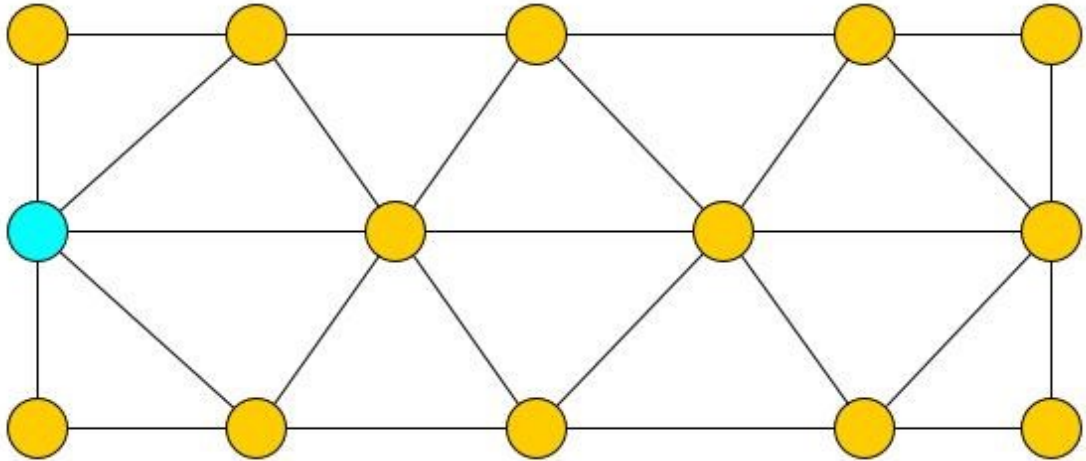
Obrázek 16: Graf pro ukázkou Fleuryho algoritmu

Nejprve musíme zjistit, jestli Eulerův tah v grafu vůbec existuje. K tomu budeme potřebovat spočítat stupeň každého vrcholu, to znamená spočítat kolik hran vychází z každého vrcholu viz Obrázek 17.



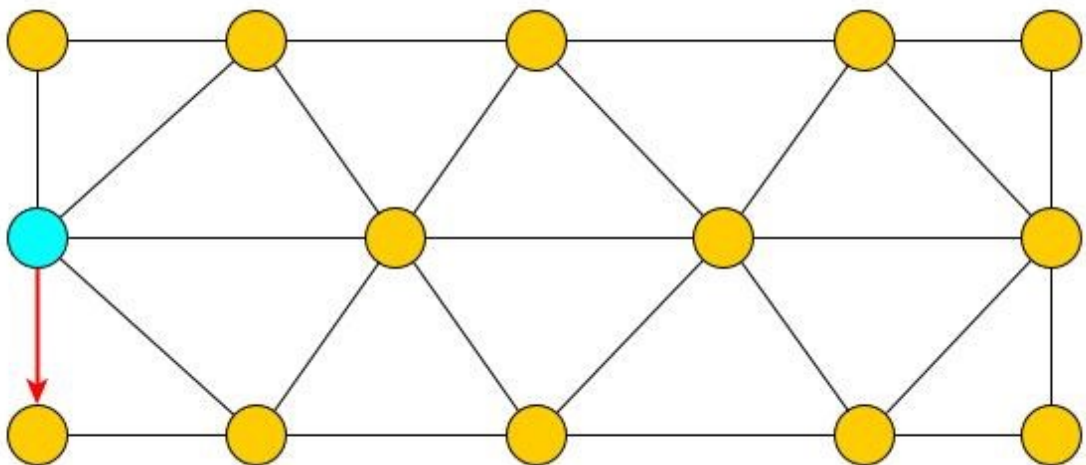
Obrázek 17: Určení stupně vrcholů

Po určení stupně všech vrcholů můžeme vidět, že náš graf obsahuje dva vrcholy lichého stupně a to stupně 5 a zároveň všechny ostatní vrcholy jsou sudého stupně, a to buď stupně 2 anebo 4. To znamená že graf obsahuje otevřený eulerovský tah. Jelikož se jedná o otevřený eulerovský tah, zvolíme počáteční vrchol takový, který má lichý stupeň v našem případě 5, neboli vytvoříme si pomocnou hranu mezi dvěma vrcholy lichého stupně a touto hranou projdeme. Kdyby byly všechny vrcholy stupně lichého, můžeme si za počátek tahu zvolit kterýkoliv vrchol viz Obrázek 18.



Obrázek 18: Zvolení počátečního vrcholu

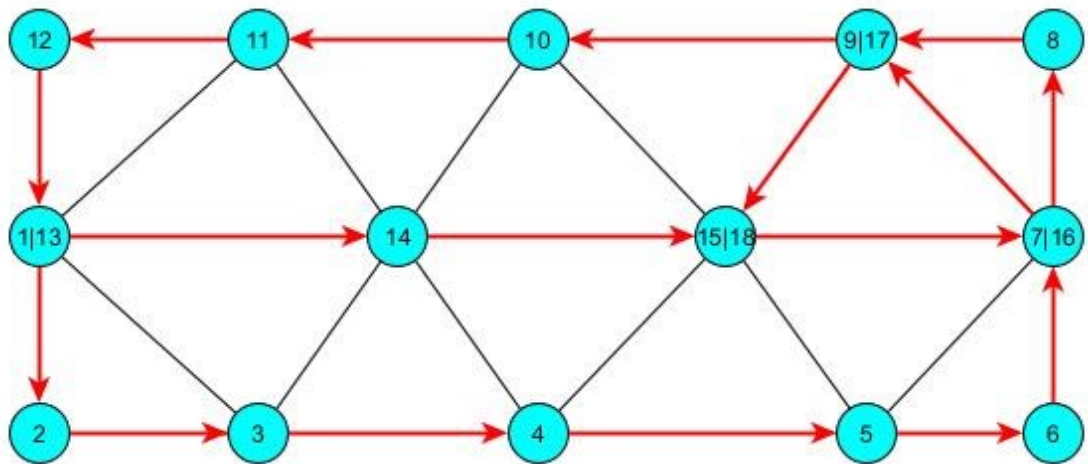
Pokračujeme na další krok algoritmu, zvolíme takovou hranu, která vychází z počátečního vrcholu, co není most, to znamená že se po jejím odebrání graf nerozpadne na více komponent. V tomto případě ani jedna z 5 hran není mostem, proto můžeme zvolit kteroukoliv z nich. Následně se přesuneme na koncový bod hrany a hranu z grafu označíme jako navštívenou (označíme červeně). Dále zkontrolujeme, jestli nám ještě zbývají nějaké nenavštívené hrany, pokud ano přesuneme se znovu na krok 3 Obrázek 19.



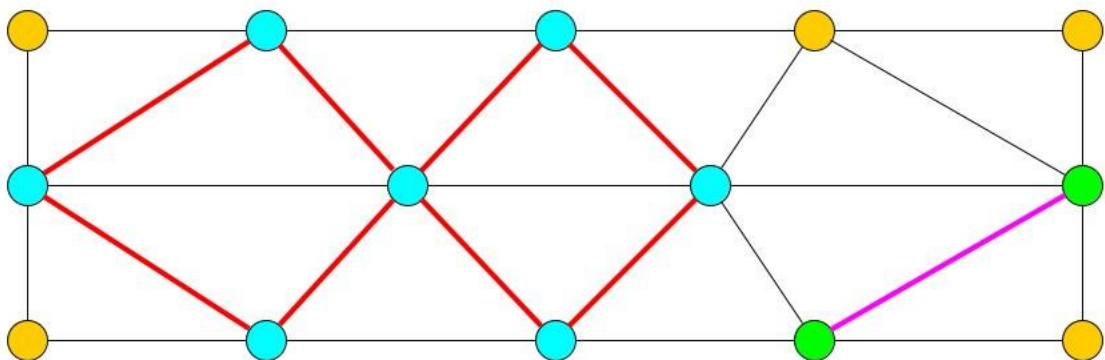
Obrázek 19: Vybrání hrany

Je velice důležité nejdříve vybírat hrany, co nejsou mosty. V následující situaci jsme se ocitli v situaci, kde máme na výběr mezi dvěma hranami, co nejsou mosty a mezi hranou, která vede do vrcholu s pořadovým číslem 5 viz Obrázek 20. Tato hrana je mostem, protože pokud bychom

ji označili jako navštívenou, graf by se nám rozdělil na dvě komponenty souvislosti, a tím pádem by nebylo možné Eulerův tah dokončit viz Obrázek 21.

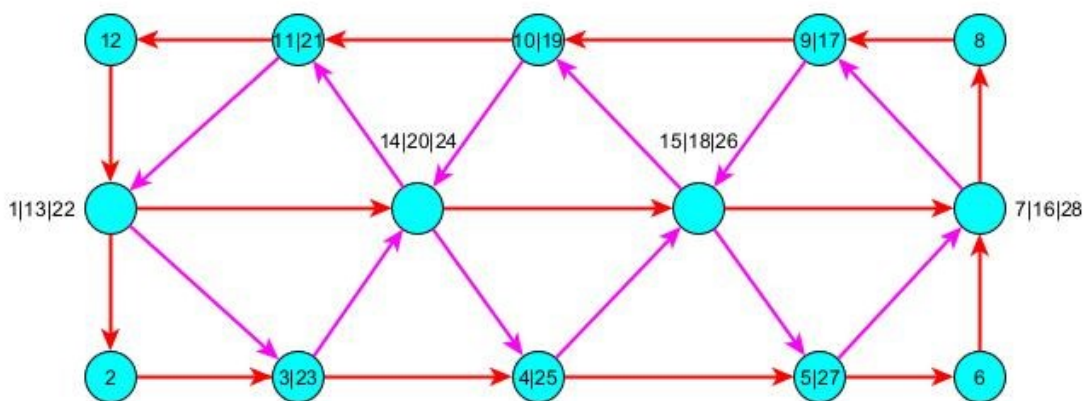


Obrázek 20: Vybírání hran, co nejsou mosty



Obrázek 21: Dvě komponenty souvislosti

Na obrázku dole můžeme vidět jedno z možných řešení Eulerova tahu, můžeme si všimnout, že tah začíná ve vrcholu s lichým stupněm a končí v druhém vrcholu s lichým stupněm. Pokud by graf měl všechny vrcholy sudého stupně, začínal by i končil ve stejném vrcholu viz Obrázek 22.



Obrázek 22: Možné řešení Eulerova tahu

2.1.5 Aplikace Eulerova tahu

Jak již bylo zmíněno jednou z nejzákladnějších hlavolamů využívající Eulerův tah je kreslení obrázku jedním tahem, tak aby se hrany neopakovaly. Tento hlavolam má řešení, pokud daný obrázek překreslíme do grafu, a tento graf obsahuje otevřený nebo uzavřený Eulerův tah.[17]

Dalším problémem využívající Eulerův tah je problém čínského pošťáka, která spočívá v tom, že pošťák musí doručit poštu do všech zadaných ulic. Zároveň musí projít všemi ulicemi tak, aby byla obchůzka co nejméně namáhavá, to znamená, aby dráha, kterou při obchůzce urazí, byla co nejkratší. Úloha se převede na neorientovaný graf tak, že vrcholy jsou křižovatky a hrany jsou ulice, které pošťák musí obejít. Hrany jsou ohodnoceny délkou dané ulice. Jestliže graf obsahuje Eulerův tah, pak je právě Eulerův tah nejkratším možným řešením, jinými slovy pošťák projde každou ulicí právě jednou. Jestliže eulerovský tah neexistuje, musí pošťák některou ulicí projít dvakrát. To znamená, že hledáme zdvojení některých hran, tak aby nový vzniklý graf byl eulerovský a aby součet hran, které jsme přidali, byl co nejmenší. [17]

Eulerovský tah má mnohé využití, zejména proto, že je základem optimálního hledání cesty. Hledání cest šlo za poslední roky hodně kupředu vyvíjením nových algoritmů. Jedním z nejpopulárnějších algoritmů, který využívá Eulerův tah jsou Google mapy a jejich efektivní algoritmus na hledání tras. Nebo například řidiči Uberu využívají Eulerův tah k hledání zákazníků, dále například Tesla a její autopilot. Pošťáci, jak již bylo zmíněno využívají Eulerův tah k tomu, aby nemuseli procházet danou ulici víckrát, než je potřeba. V širším měřítku Eulerův tah využívají například malíři, popeláři, piloti, vývojáři GPS, nebo obchodníci distribuující reklamu, zkrátka každý, kdo používá jakékoli cesty. [15]

2.2 Hamiltonová kružnice

Při hledání eulerovského tahu v daném grafu, jsme měli za úkol navštívit každou hranu pouze jednou, a při tom nezáleželo na tom, kolikrát navštívíme každý vrchol. Analogicky můžeme formulovat podobnou úlohu, ve které nemusíme navštívit všechny hrany, ale musíme navštívit každý vrchol pouze jednou. Tato úloha je motivována problémem obchodního cestujícího, který má za úkol navštívit všechna zadaná města právě jednou a navrátit se do města, ve kterém svoji cestu začínal. Terminologií teorie grafů můžeme úlohu formulovat tak, že máme za úkol najít takovou kružnici v grafu, který obsahuje všechny vrcholy daného grafu. Takováto kružnice ale nemusí nutně existovat v každém grafu. [2]

2.2.1 Hamiltonovský graf

Hamiltonovský graf byl zaveden jako analogie eulerovského grafu. Na první pohled se může hledání hamiltonovské kružnice jevit jako velmi podobné eulerovskému tahu. Ale zatímco existence eulerovského tahu v souvislém grafu má nutnou i postačující podmínku, a to počet lichých vrcholů grafu, existence hamiltonovské kružnice v grafu žádnou takto jednoduše ověřitelnou podmínku nemá. [2]

2.2.2 Postačující podmínky existence hamiltonovské kružnice

Pro obecný graf je těžké rozhodnout, jestli obsahuje hamiltonovskou kružnici. Existuje ale alespoň několik jednoduchých postačujících podmínek, které když graf splňuje, tak můžeme říct, že obsahuje hamiltonovskou kružnici. [2]

Jedna z prvních postačujících podmínek existence hamiltonovské kružnice v neorientovaném grafu pochází z roku 1960, kdy ji formuloval Oystein Ore. [19]

Věta 2.1 (Oreho věta): *Mějme graf G s n vrcholy, kde $n \geq 3$. Jestliže pro každé dva nesousední vrcholy u, v grafu G platí $\deg(u) + \deg(v) \geq n$, tak graf G je hamiltonovský.* [19]

Speciální případ této věty dokázal v roce 1952 dokázal evropský matematik Gabriel Andrew Dirac. [19]

Věta 2.2 (Diracova věta): *Mějme graf G s n vrcholy, kde $n \geq 3$. Je-li $\delta(G) \geq n/2$, tak graf G je hamiltonovský.* [19]

Diracova věta je důsledek Oreho věty. V roce 1962 dokázal maďarský matematik Lajos Pósa ještě obecnější větu. [19]

Věta 2.3 (Pósova věta): *Mějme graf G s n vrcholy, kde $n \geq 3$. Jestliže pro každé přirozené číslo $j < n/2$ obsahuje graf G méně než j vrcholů stupně menšího nebo rovného j , tak graf G je hamiltonovský.* [19]

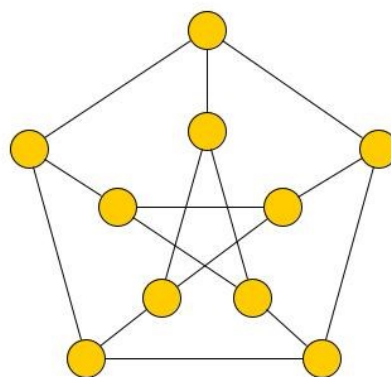
2.2.3 Nutné podmínky existence hamiltonovské kružnice

Každý hamiltonovský graf je souvislý, dokonce platí, že každý hamiltonovský graf je alespoň 2-souvislý, to znamená, že neobsahuje artikulaci. [19]

Věta 2.4: *Je-li G hamiltonovský graf, tak pro každou neprázdnou vlastní podmnožinu S vrcholové množiny $V(G)$ platí $\omega(G - S) \leq |S|$.* [19]

Jinými slovy lze říct, že z každé komponenty grafu vede hrana do jiného vrcholu množiny S , to znamená, že množina S obsahuje alespoň tolik vrcholů, kolik je v grafu komponent $G - S$. [19]

Postačující podmínky se používají k tomu, abychom ukázali, že je graf hamiltonovský. Nutné podmínky se používají v k ukázání, že graf hamiltonovský není. Pokud graf nějakou z nutných podmínek nesplňuje, můžeme říct, že hamiltonovský není. Zároveň je ale možné, že graf všechny nutné podmínky splňuje, a přesto hamiltonovský není. Takovým grafem je například Petersenův graf. [19]



Obrázek 23: Petersenův graf (inspirováno zdrojem [18])

2.2.4 Algoritmus pro hledání hamiltonovské kružnice

Rozhodnout, zda je daný graf hamiltonovský je složitý problém. Tento problém patří k takzvaným NP-úplným problémům. NP-úplné problémy jsou matematické problémy, které jsou považovány za jedny z nejsložitějších problémů. Jejich časová složitost je exponenciální, nebo v horším případě faktoriálová. [18]

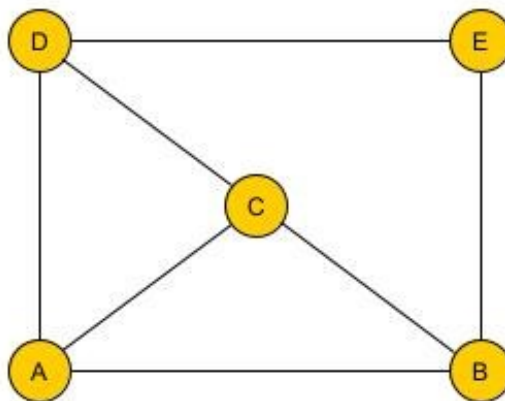
Jedním z algoritmů pro hledání hamiltonovské kružnice je takzvaný backtrack algoritmus. Tento algoritmus vytváří postupně kružnici vrchol za vrcholem. Začne v libovolném vrcholu a následně se snaží prodloužit cestu jedním z nenavštívených sousedů aktuálního vrcholu. Pokud už není možné cestu prodloužit a kružnice není hotova, algoritmus „backtrackuje“ na předchozí vrchol tak, že odstraní aktuální vrchol z cesty. [26]

Algoritmus:

- 1) Zvolíme počáteční vrchol a přidáme ho do cesty.
- 2) Přidáme do cesty některý z nenavštívených sousedů posledního vrcholu v cestě.
- 3) Pokud byly všechny vrcholy navštíveny a sousední vrchol posledního vrcholu v cestě je počáteční vrchol, algoritmus končí, hamiltonovská kružnice byla nalezena.
- 4) Pokud není možné cestu prodloužit, všechny vrcholy posledního vrcholu v cestě byly navštíveny, odstraníme poslední vrchol z cesty. [18]

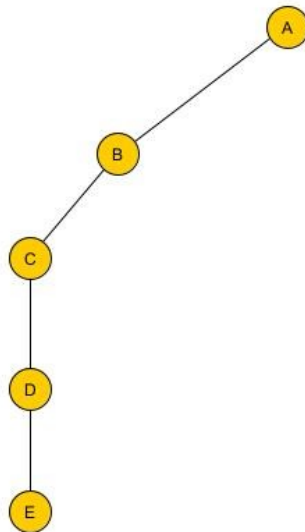
2.2.5 Příklad backtracking algoritmu

Backtracking algoritmus si ukážeme na následujícím grafu G viz Obrázek 24.



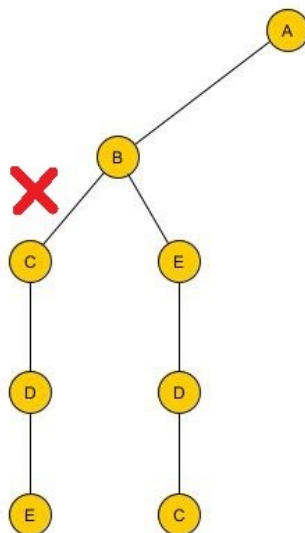
Obrázek 24: Graf G pro ukázkou backtracking algoritmu

Zvolíme počáteční vrchol například vrchol A, a přidáme ho do cesty. Následně zvolíme jeden z jeho nenavštívených vrcholů například vrchol B a také ho přidáme do cesty. Pokračujeme stejným způsobem dále přidáme vrchol C vrchol D a vrchol E do cesty *Obrázek 25*.



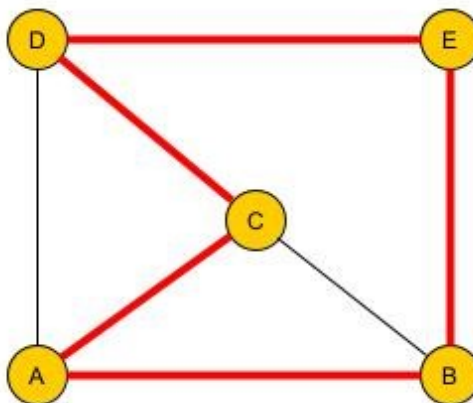
Obrázek 25: Ukázka backtracking algoritmu č. 1

Jak můžeme vidět, navštívili jsme všechny vrcholy, ale počáteční vrchol A není sousedním vrcholem vrcholu E. Proto postupně odstraňujeme vrcholy, které nemají žádné jiné nenavštívené sousedy, a to vrcholy E, D a C. Po backtrackování na vrchol B zjistíme, že vrchol B má ještě jednoho nenavštíveného souseda a to vrchol E. Přidáme ho tedy do cesty a následně opět postupně přidáváme jeho nenavštívené sousedy, a to vrcholy D a C viz *Obrázek 26*.



Obrázek 26: Ukázka backtracking algoritmu č. 2

Jak můžeme vidět, opět jsme navštívili všechny vrcholy. V tomto případě ale poslední vrchol C je sousedním vrcholem počátečního vrcholu A. Nalezli jsme tedy hamiltonovskou kružnici A, B, E, D, C, A viz *Obrázek 27*.



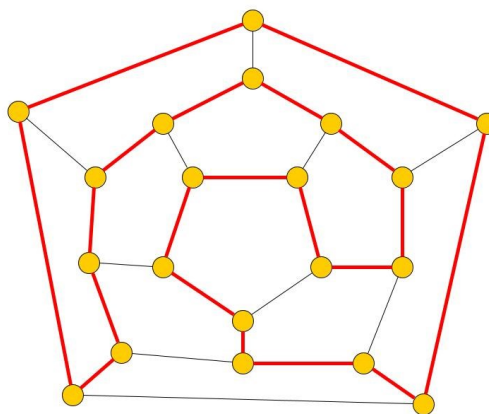
Obrázek 27: Hamiltonovská kružnice v grafu G

2.2.6 Icosian game

Hamiltonovské grafy jsou pojmenovány podle irského matematika, astronoma a fyzika Williama Rowana Hamiltona, který v roce 1856 našel model cest v grafu pravidelného dvanáctistěnu, který nazval *The Icosian Calculus*. Tento model následně prodal výrobci a od roku 1859 se tato hra objevila na trhu pod názvem *The Icosian Game*, jejíž cílem bylo natáhnout provázek, na kolíky dvanáctistěnu tak, aby vytvořilo kružnici viz *Obrázek 28*. Tato hra se stala velmi populární, a proto dostal tento typ grafu název hamiltonovská kružnice viz *Obrázek 29*. [18]



Obrázek 28: The Icosian Game [20]



Obrázek 29: Hamiltonovská kružnice ve dvanáctistěnu (inspirováno zdrojem [18])

2.2.7 Obchodní cestující

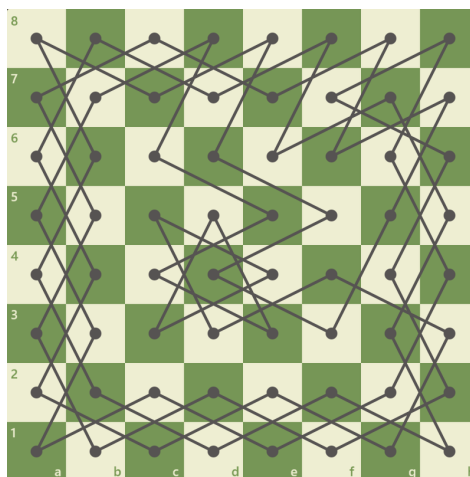
Problém obchodního cestujícího spočívá v tom, že musí navštívit všechna zadaná města, při tom musí každé město navštívit pouze jednou a musí skončit ve stejném městě, ve kterém začal. To znamená, že hledáme hamiltonovskou kružnici. Navíc je nutné, aby dráha, kterou obchodní cestující ujede, byla co nejmenší. Jinými slovy, hledáme minimální možnou kružnici v grafu s ohodnocenými hranami. To znamená, že hledáme minimální hamiltonovskou kružnici v ohodnoceném grafu. [18]

2.2.8 Jezdcova procházka

Další známou hádankou popisující hamiltonovskou kružnici, je takzvaná jezdcova procházka (anglicky Knight's tour). Jedná se o problém týkající se šachu, kde jezdec představuje klasickou šachovou figurku jezdece (koně), která se pohybuje ve skocích ve tvaru písmene „L“. [18]

Tento šachový problém se poprvé objevil už v 9. století. Problém spočívá v tom, že jezdec, který začíná na kterémkoliv políčku šachovnice, musí postupně skočit na ostatních 63 políček šachovnice, tak aby na stejné políčko nestoupil vícekrát než jednou. To znamená najít hamiltonovskou kružnici. Tento problém lze vyřešit dvěma různými způsoby, jedním způsobem je otevřená procházka a druhým způsobem je uzavřená procházka. Jinými slovy hamiltonovská cesta a hamiltonovská kružnice. Při otevřené procházce projde jezdec každým políčkem na šachovnici právě jednou. U uzavřené procházky projde každým políčkem právě jednou s výjimkou startovního pole, na které se vrátí po dokončení trasy. Proto je uzavřená procházka o jeden skok delší než otevřená procházka a nazývá se hamiltonovskou kružnicí. Platí, že každá hamiltonovská kružnice obsahuje hamiltonovskou cestu. Toto tvrzení ale neplatí

obráceně, to znamená, že pokud prodloužíme hamiltonovskou cestu, nemusí to mít za výsledek vytvoření hamiltonovské kružnice. Na začátku 18. století De Montmort a Abraham De Moivre vyřešili tento problém poprvé. Moivrovo řešení spočívalo v tom, že jezdec skákal po okraji šachovnice, dokud to bylo možné, a když to nešlo, provedl skok do středového čtverce. Leonhard Euler v polovině 18. století podrobněji vyřešil tento problém a dokázal, že není možné najít řešení tohoto problému na šachovnicích s lichými rozměry $N \times N$. [18][21]



Obrázek 30: Příklad uzavřené jezdcovi procházky [21]

2.3 Rovinné zobrazení grafu

Rovinné zobrazení grafu je takové zobrazení, kde je graf vyobrazen tak, že žádné dvě jeho hrany se neprotínají, jak bylo již vysvětleno v kapitole 1.3.10. Toto zobrazení je velmi důležité v mnoha praktických aplikacích, například při návrhu elektrických obvodů a tištěných spojů. V těchto případech je žádoucí minimalizovat počet křížení hran, protože tyto křížení představují křížení spojů, které musí být přemostěny. [19]

Definice 2.1: *Rovinným nakreslením grafu G je takové nakreslení grafu, ve kterém jsou vrcholy znázorněny jako různé body v rovině a hrany jako křivky spojující tyto body, které odpovídají koncovým vrcholům. Přitom žádná hrana se nesmí křížit ani procházet jinými body, než které odpovídají jejím koncovým vrcholům. Řekneme, že graf je rovinný, pokud máme jeho rovinné nakreslení.* [19]

Grafy, které mají rovinné zobrazení, tedy lze je nakreslit tak, aby se žádné dvě hrany grafu nekřížily, se nazývají rovinné nebo jsou také někdy označovány jako planární. [19]

Při studiu rovinných grafů je důležité zaměřit se nejen na vrcholy a hrany, ale také na oblasti, které vzniknou rozdělením roviny podle daného nakreslení grafu. V roce 1750 si Leonhard

Euler povšiml, že existuje určitý vztah mezi počtem oblastí, hran a vrcholů rovinného grafu. [19]

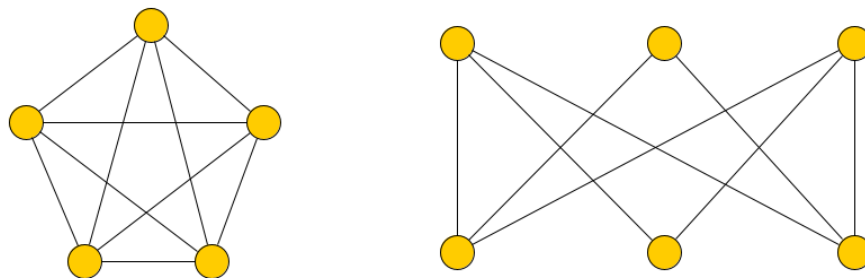
Věta 2.5 (Eulerův vzorec): *Mějme jednoduchý souvislý rovinný graf s f oblastmi, v vrcholy a h hranami. Potom platí $v + f - h = 2$. [19]*

Tento vztah má mnoho důsledků mezi ně patří:

Důsledek 2.1: *Jednoduchý rovinný graf na v vrcholech, kde $v \geq 3$, má nejvýše $3v - 6$ hran. [19]*

Důsledek 2.2: *Jednoduchý rovinný graf bez trojúhelníků C_3 na v vrcholech, kde $v \geq 3$, má nejvýše $2v - 4$ hran. [19]*

„Podle těchto důsledků můžeme určit, že grafy K_5 a $K_{3,3}$ nejsou rovinné viz Obrázek 31. Graf K_5 má 5 vrcholů a 10 hran, graf obsahující 5 vrcholů musí mít nejvýše $3 \times 5 - 6 = 9$ hran. Graf K_5 jich má však 10, proto není rovinný. Graf $K_{3,3}$ má 6 vrcholů a 9 hran, graf obsahující 6 vrcholů, a navíc neobsahující žádné trojúhelníky musí mít nejvýše $2 \times 6 - 4 = 8$ hran. Graf $K_{3,3}$ jich má však 9, proto není rovinný.“ [19]



Obrázek 31: Nerovinné grafy K_5 a $K_{3,3}$ (inspirováno zdrojem [19])

Definice 2.2: Dělení grafu G rozumíme graf, který vznikne posloupností operace **dělení hrany**. Je-li $G = (V, X, p)$ graf s hranou $h \in X$, $p(h) = (x, y)$ a vrchol $z \in V$, vznikne dělením hrany h graf $G' = (V', X', p')$, pro který platí $V' = V \cup \{z\}$, $X' = (X \setminus \{h\}) \cup \{h_1, h_2\}$, kde $p'(h_1) = (x, z)$ a $p'(h_2) = (z, y)$. Opačná operace k dělení hrany se nazývá **vypouštění vrcholu**. [27]

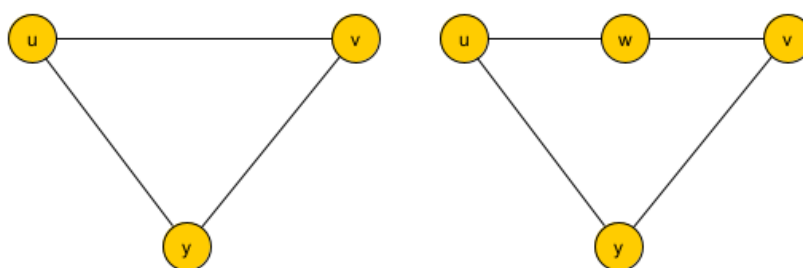
Definice 2.3: Řekneme, že graf G_1 je **Homeomorfní** s grafem G_2 pokud:

- (1) je s grafem G_2 izomorfni,
- (2) je možné konečným dělením hran, nebo vypouštěním vrcholů dosáhnout toho, že vzniklé grafy jsou izomorfni. [27]

V roce 1930 Kazimierz Kuratowski ukázal, že graf K_5 a $K_{3,3}$ mají klíčové postavení. Jejich struktura neumožňuje rovinné nakreslení, a zároveň žádná jiná takováto struktura neexistuje. Kuratowski zavedl následující tvrzení. [19]

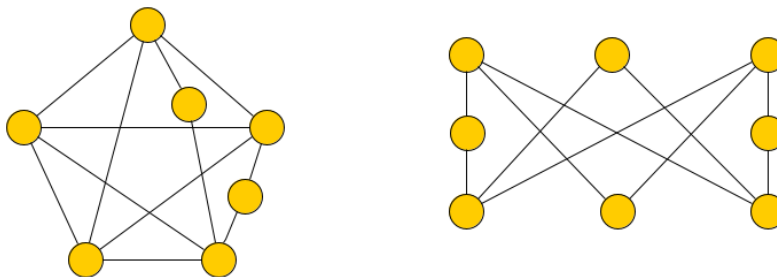
Věta 2.6 *Graf G je rovinný právě když, žádný jeho podgraf není homeomorfní s grafem K_5 , nebo $K_{3,3}$.* [19]

Dělení hran si uvedeme na příkladu. Máme graf G , do kterého přidáme nový vrchol w a odebereme nějakou hranu z grafu například uv a následně do grafu přidáme hrany uw a wv . Nově vzniklý graf je dělením grafu G viz Obrázek 32. [19]



Obrázek 32: Graf G a jeho rozdělení vzniklé půlením hran

Dělení grafů K_5 a $K_{3,3}$ existuje nekonečně mnoho příklady rozdělení grafů K_5 a $K_{3,3}$ jsou například tyto viz Obrázek 33. [19]



Obrázek 33: Příklady homeomorfních grafů K_5 a $K_{3,3}$ (inspirováno zdrojem [19])

2.4 Barvení rovinného grafu – problém čtyř barev

Existuje spojitost mezi problémem barvení grafu a problémem barvení politických map. Představme si politickou mapu Evropy viz Obrázek 34. [3]



Obrázek 34: Politická mapa Evropy [3]

Předpokládáme, že každý stát je souvislá oblast, která je ohraničena nějakou topologickou kružnicí (z tohoto důvodu jsou vynechány státy jako Británie, Irsko nebo Rusko). Dva státy jsou sousední, pokud mají společnou aspoň malou část hranice. Státy, jejichž hranice se dotýká jen v jednom, nebo několika bodech nepovažujeme za sousední. Problém spočívá v tom, že je potřeba každému státu na této mapě přiřadit nějakou barvu tak, aby dva státy, které jsou mezi sebou sousedící nebyly obarveny stejnou barvou. K obarvení této mapy stačí čtyři barvy. Otázka, zda k obarvení každé politické mapy stačí čtyřmi barvy je jeden z významných kombinatorických problémů. [3]

Tento problém se nazývá problém čtyř barev. Tento problém se snažilo vyřešit hodně matematiků více než sto let, ale tento problém byl vyřešen až v roce 1976. [22]

2.4.1 Problém v teorii grafů

Problém čtyř barev se dá analogicky převést na barvení grafu. Každý stát zobrazíme jako vrchol grafu. Hrany každého vrcholu budou hranice mezi dvěma státy. Cílem je tedy obarvit všechny vrcholy grafu tak, aby žádné dva vrcholy spojené hranou nebyly obarveny stejnou barvou. [22]

2.4.2 Historie problému čtyř barev

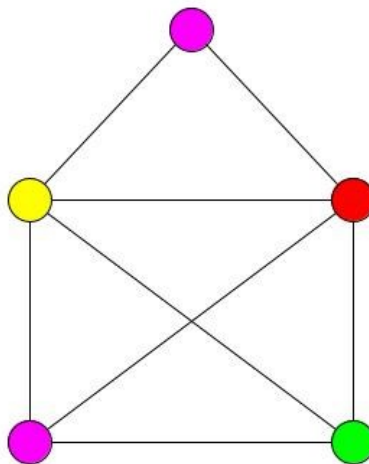
Problém čtyř barev byl poprvé zmíněn v roce 1840 matematikem A. F. Mobiusem. Později se o něm zmiňuje A. de Morgan ve svém dopise W. R. Hamiltonovi, kde popisuje práci jeho

studenta Frederica Guthrie, který našel způsob, jak obarvit mapu Anglie čtyřmi barvami. Tento problém také zkoumal A. Cayley v roce 1878 a C. S. Peirce v roce 1860 se pokusil tento problém matematicky dokázat, ale neuspěl. Další matematici se pokoušeli tento problém dokázat, ale stále neúspěšně. V roce 1879 publikoval A. B. Kempe důkaz problému čtyř barev, ale v roce 1890 byl tento důkaz vyvrácen J. P. Heawodem, který ho upravil a dokázal, že každou politickou mapu lze obarvit pěti barvami. Teprve v roce 1976 se podařilo formulovat důkaz problému čtyř barev matematikům K. Appelovi a W. Hakenovi z Univerzity v Illinois. Tento důkaz zabral 1200 hodin procesorového času. Právě z tohoto důvodu není jednoznačně přijímán, ale přesto nebyl dodnes vyvrácen. [23]

2.4.3 Barvení grafu

Nejběžnější typ barvení grafu je vrcholové barvení grafu. Příklad vrcholového obarvení grafu může být následující viz Obrázek 35. [23]

Definice 2.4: *Obarvení grafu* G pomocí k barev je takové zobrazení $c: V(G) \rightarrow \{1, 2, \dots, k\}$, ve kterém každé dva vrcholy, které jsou spojené hranou, budou mít různou barvu, tj. $c(u) \neq c(v)$ pro každou hranu $uv \in E(G)$. Uvedenému obarvení vrcholů grafu se říká také dobré vrcholové barvení grafu. [2]



Obrázek 35: Příklad vrcholového barvení grafu

Pro každý rovinný graf platí následující věta:

Věta 2.7: *Každý rovinný graf lze obarvit nejvýše čtyřmi barvami.* [2]

2.4.4 Algoritmus pro barvení grafu

Problém barvení grafu se řadí mezi NP-úplné problémy, proto je algoritmizace velmi složitá. Většina algoritmů, které se používají k hledání obarvení grafu, nehledají optimální řešení, ale jedno z možných řešení. [23]

Jedním z možných a nejjednodušších algoritmů je náhodné barvení grafu. Tento algoritmus funguje na principu generování náhodných obarvení grafu, každé takto vygenerované obarvení je testováno, zda splňuje podmínky takzvaného dobrého obarvení grafu. Toto řešení je však většinou v praxi nepoužitelné. [23]

Dalším možným algoritmem je takzvané sekvenční barvení nebo také hladové z anglického slova „greedy“. Tento algoritmus spočívá v postupném barvení vrcholů grafu tak, že barva pro obarvení daného vrcholu je vždy volena tak, aby splňovala podmínku dobrého barvení grafu. Vrchol, kterému je přiřazena barva je definitivně obarven, to znamená, že přiřazenou barvu vrcholu už nelze změnit. Důležité u tohoto algoritmu je volba vrcholu, který budeme barvit. To lze udělat několika způsoby: [23]

- Náhodný výběr vrcholů, také označován jako First Fit (FF), je nejjednodušší přístup, kde volba následující vrcholu k obarvení je čistě náhodná. [23]
- Sestupné řazení vrcholu, také označován jako Large Degree Ordering (LDO), spočívá v tom, že jako vrchol k obarvení vždy vybereme vrchol s nejvyšším stupněm, pokud mají dva vrcholy stejný stupeň vybereme jeden z nich. Tento výběr zajistí, že nejprve budou obarveny vrcholy, pro které je obarvení nejsložitější. [23]

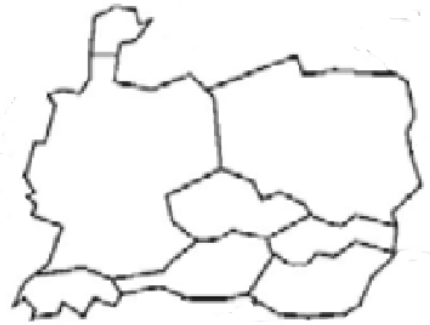
Algoritmus sekvenčního barvení se sestupným barvením vrcholu:

- (1) Nejdříve určíme stupeň všech vrcholů a barvám k obarvení grafu přiřadíme čísla 1 až 4.
- (2) Vybereme takový neobarvený vrchol, který má nejvyšší stupeň, v případě že je více neobarvených vrcholů se stejným stupněm vybereme jeden z nich. Do seznamu možných barev přidáme všechny čtyři barvy.
- (3) Postupně procházíme všechny sousední obarvené vrcholy aktuálně barveného vrcholu, barvy těchto vrcholů odstraníme ze seznamu možných barev a následně obarvíme aktuální vrchol barvou s nejmenším číslem ze seznamu možných barev.

- (4) Pokud jsou všechny vrcholy obarvené algoritmus končí, pokud ne vrátíme se na krok číslo 2. [28]

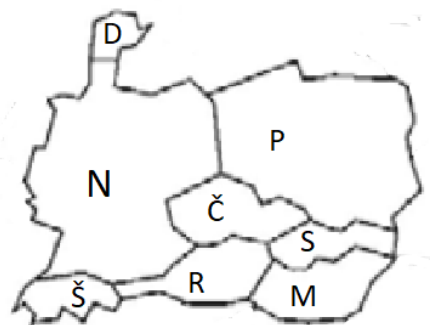
2.4.5 Ukázka sekvenčního barvení

Sekvenční barvení si ukážeme na již zmíněném problému barvení politické mapy viz Obrázek 34. Pro zjednodušení budeme barvit pouze část mapy (střední Evropu).



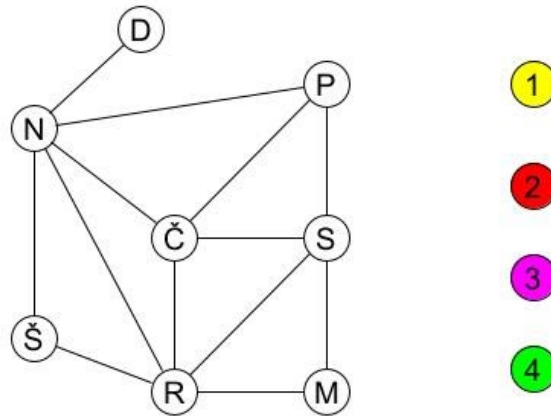
Obrázek 36: Mapa střední Evropy[3]

Nejprve každému státu přiřadíme počáteční písmeno daného státu, které bude sloužit jako označení vrcholu viz Obrázek 37.



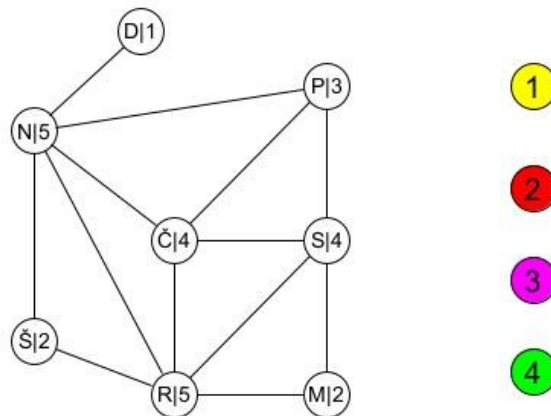
Obrázek 37: Popsání států[3]

Následně převedeme mapu na graf tak, že státy představují jednotlivé vrcholy grafu, hranice mezi nimi představují hrany grafu a popis vrcholů odpovídá písmenům přiřazeným v předchozím kroku a zároveň si zvolíme 4 barvy, kterými budeme graf barvit a přiřadíme jim čísla 1 až 4 viz Obrázek 38.



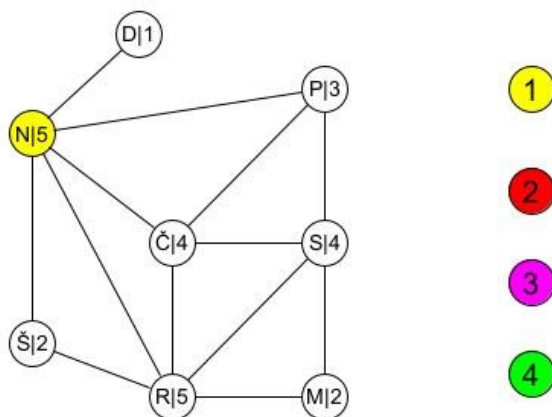
Obrázek 38: Převedení mapy do grafu

Následně určíme stupeň každého vrcholu tak, že spočítáme počet hran, které vedou z anebo do něj viz Obrázek 39.



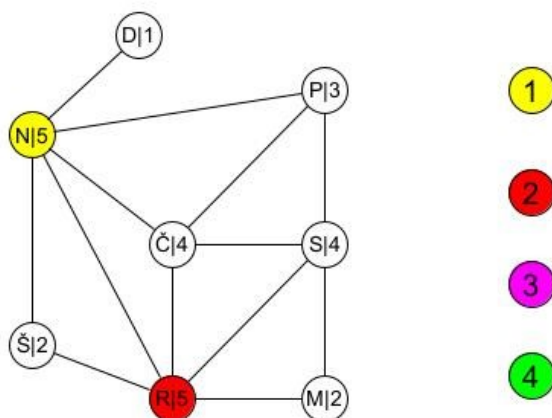
Obrázek 39: Určení stupně vrcholů

Následně vybereme neobarvený vrchol s nejvyšším stupněm, v tomto případě máme na výběr z vrcholu N a vrcholu R se stupněm 5. Vybereme jeden z nich například vrchol N. Jelikož nemá žádné obarvené sousedy zvolíme barvu s nejnižším číslem a to číslem 1 a vrchol touto barvou obarvíme.



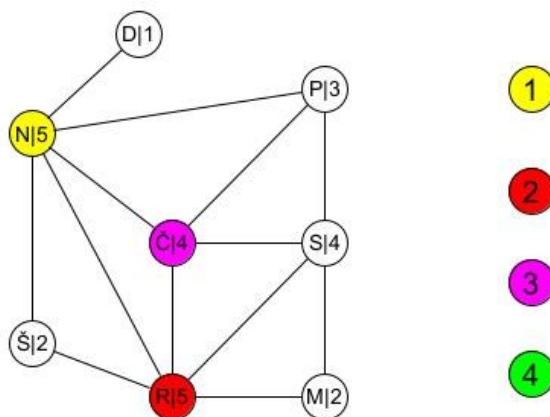
Obrázek 40: Obarvení vrcholu N

Tento postup opakujeme, opět zvolíme neobarvený vrchol s nejvyšším stupněm, a to vrchol R se stupněm 5. Procházíme jeho obarvené sousedy a zjistíme, že vrchol R sousedí s vrcholem obarveným barvou s číslem 1, proto tuto barvu nemůžeme použít. Vybereme tedy barvu s nejnižším číslem z barev, které zbyly, a to barvu s číslem 2 viz Obrázek 41.



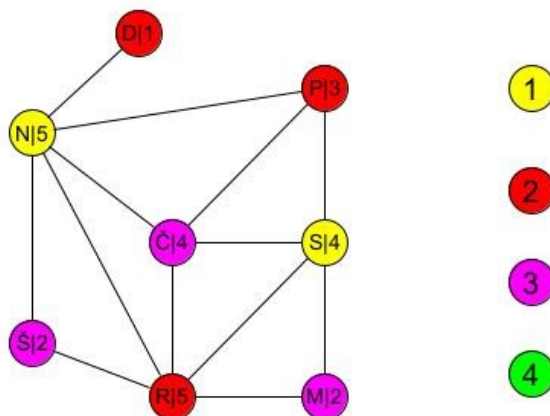
Obrázek 41: Obarvení vrcholu R

Zvolíme další neobarvený vrchol s nejvyšším stupněm, a to například vrchol Č, procházíme jeho obarvené sousedy a odebereme barvy 1 a 2. A opět zvolíme barvu s nejnižším číslem a to barvu 3 viz Obrázek 42.



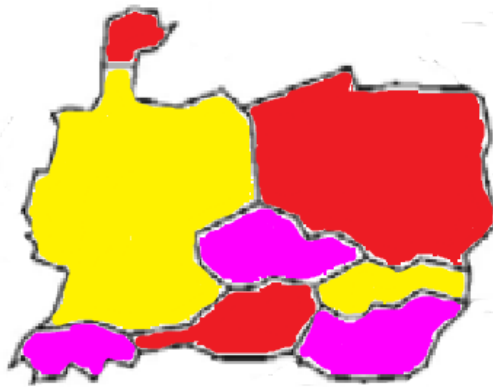
Obrázek 42: Obarvení vrcholu Č

Toto stále opakujeme, dokud nejsou všechny vrcholy obarveny, výsledné obarvení grafu vypadá takto viz Obrázek 43.



Obrázek 43: Výsledné obarvení grafu

Následně vybarvíme mapu tak, že každému státu přiřadíme stejnou barvu, jakou má vrchol s odpovídajícím písmenem. Výsledné obarvení mapy vypadá takto viz Obrázek 44. Můžeme vidět, že žádné dva sousední státy nemají stejnou barvu.



Obrázek 44: Obarvená mapa (zdroj původního obrázku: [3])

2.4.6 Další aplikace barvení grafu

Barvení grafů je široce využívaná metoda v matematice, informatice i reálném životě. Jedním z nejnámějších a nejčastějších problémů, které využívají barvení grafů, je právě problém barvení politických map. V tomto případě jsou vrcholy grafu přiřazeny jednotlivým státům a hrany představují hranice mezi nimi. Cílem je obarvit jednotlivé státy tak, aby sousedící státy měly různou barvu. [23]

Dalším příkladem z praxe, kde můžeme využít barvení grafů, je skladování nebezpečných látek. Tento problém spočívá v tom, jak uskladnit velké množství různých nebezpečných látek tak, aby byly v bezpečných podmínkách a aby se předešlo kombinacím, které by byly nebezpečné. Vrcholy grafu představují jednotlivé nebezpečné látky a hrany mezi nimi reprezentují kombinace nebezpečných látek, které nesmějí být uskladněny společně. Barva vrcholu určuje prostor, kde může být daná látka uskladněna. [23]

Jedním z dalších praktických využití barvení grafů je plánování procesů. Barvení grafů může být využito pro plánování projektů, kde jsou některé fáze projektu vzájemně závislé a nemohou probíhat současně. Vrcholy grafu v tomto případě představují jednotlivé fáze projektu a hrany mezi nimi označují vzájemnou závislost. Barva vrcholu určuje časový úsek, ve kterém může daná fáze projektu být dokončena. Tento přístup k plánování projektů umožňuje zefektivnit celkový průběh projektu a minimalizovat zpoždění. [23]

Dále lze barvení grafů využít například při tvorbě rozvrhu, kdy se nesmí dvě hodiny křížit, nebo například v divadelním představení, kde vystupuje více postav, než má divadlo herců a ve hře není scéna, ve které by vystupovali všechny postavy najednou. [23]

2.5 Algoritmy k otestování souvislosti grafu

Pokud chceme otestovat, zda je graf souvislý, nebo najít jeho komponenty souvislosti, můžeme použít algoritmus k procházení daného grafu. Buď použijeme algoritmus k průchodu grafu do hloubky anebo do šířky. Na začátku zvolíme nějaký vrchol a z něj budeme graf procházet, přičemž postupně budeme navštěvovat dostupné vrcholy, a označíme je jako navštívené. Po dokončení algoritmu jsou označeny všechny vrcholy, které leží v dané komponentě souvislosti. Pokud jsou označeny všechny vrcholy daného grafu jako navštívené, je tento graf souvislý. Pokud nejsou všechny vrcholy označeny jako navštívené, tak jsme našli jednu z více komponent souvislosti daného grafu. [24]

2.5.1 Průchod grafu do hloubky

Průchod do hloubky se často označuje pod zkratkou DFS z anglického Depth First Search. [25]

Při průchodu grafem je potřeba, abychom navštívili všechny vrcholy, a zároveň každý vrchol navštívili pouze jednou. Z tohoto důvodu je potřeba zajistit, abychom dokázali rozlišit navštívené a nenavštívené vrcholy, to zajistíme například přidáním proměnné. Při průchodu do hloubky se vrcholy, které v dané chvíli nemůžeme navštívit, dočasně ukládají na zásobník. Průchod začneme v libovolném vrcholu. [25]

Algoritmus průchodu grafu do hloubky:

(1) *Počáteční nastavení:*

- *Zásobník je prázdný.*
- *Aktuální vrchol v nastavíme na počáteční vrchol.*

(2) *Navštívení aktuálního vrcholu:*

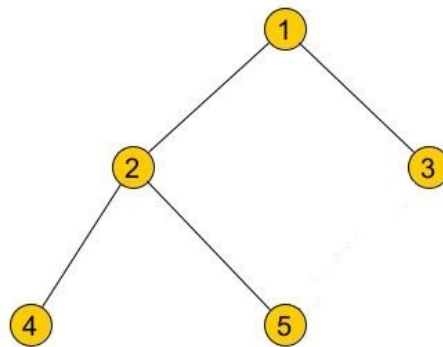
- *Označíme aktuální vrchol jako navštívený.*
- *Pokud nemá aktuální vrchol žádného nenavštíveného souseda přejdeme na krok 3.*
- *Pokud má jednoho nenavštíveného souseda nastavíme tohoto souseda jako aktuální vrchol a přejdeme opět na začátek kroku 2.*
- *Pokud má více nenavštívených sousedů, jeden z nich nastavíme jako aktuální vrchol a zbytek uložíme do zásobníku přejdeme na začátek kroku 2.*

(3) Odebrání vrcholů ze zásobníku:

- Pokud je zásobník prázdný, průchod je ukončen.
- Odebereme vrchol ze zásobníku a ověříme, zda již nebyl navštívený, a pokud ano tak opět přejdeme na začátek kroku 3.
- Pokud odebraný vrchol ještě nebyl navštívený, nastavíme ho na aktuální vrchol a přejdeme na začátek kroku 2. [25]

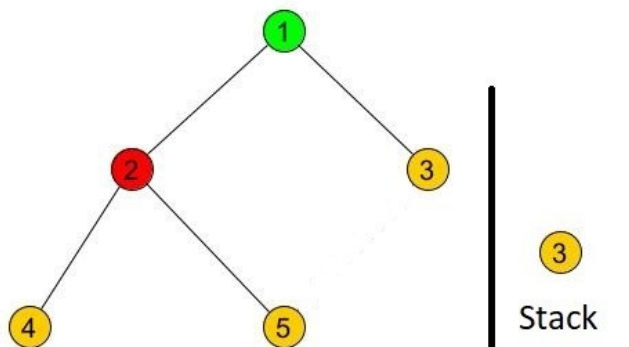
2.5.2 Příklad průchodu do hloubky

Průchod grafu do hloubky si ukážeme na následujícím grafu viz Obrázek 45.



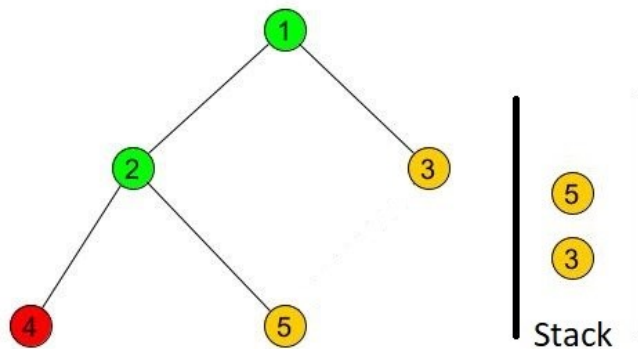
Obrázek 45: Příklad procházení grafu do hloubky (inspirováno zdrojem [25])

Jako počáteční vrchol zvolíme vrchol 1 a označíme ho za navštívený. Vrchol 1 má dva nenavštívené sousedy, proto vrchol 2 nastavíme na aktuální a vrchol 3 vložíme na zásobník. Navštívené vrcholy jsou označeny zeleně a aktuální vrchol je označen červeně viz Obrázek 46.



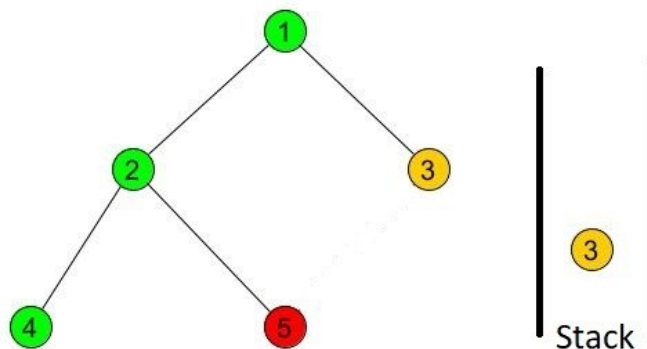
Obrázek 46: První krok prohlídky do hloubky

Aktuální vrchol, tedy vrchol 2, nastavíme jako navštívený. Vrchol 2 má opět dva nenavštívené sousedy vrcholy 4 a 5. Vrchol 4 nastavíme na aktuální a vrchol 5 vložíme do zásobníku viz Obrázek 47.



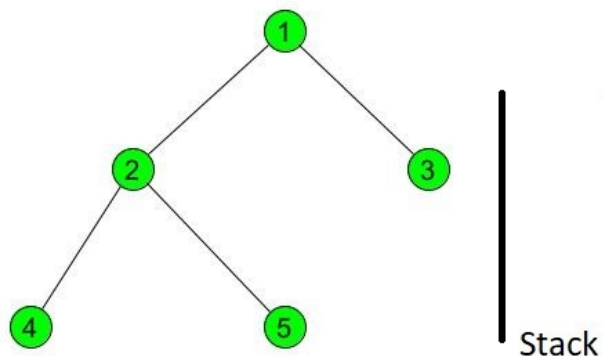
Obrázek 47: Druhý krok průchodu do hloubky

Aktuální vrchol, tedy vrchol 4, nastavíme jako navštívený. Vrchol 4 nemá žádné nenavštívené sousedy, proto přejdeme ke kroku 3 a odebereme vrchol 5 ze zásobníku a nastavíme ho jako aktuální viz Obrázek 48.



Obrázek 48: Třetí krok prohlídky do hloubky

Aktuální vrchol, tedy vrchol 5, nastavíme jako navštívený. Vrchol 5 nemá žádné nenavštívené sousedy, proto odebereme vrchol 3 ze zásobníku, nastavíme ho jako aktuální a následně jako navštívený. Vrchol 3 nemá žádné nenavštívené sousedy, proto přejdeme ke kroku 3. Zjistíme, že zásobník je prázdný, průchod tedy končí. Průchod tímto grafem do hloubky je tedy posloupnost vrcholu 1, 2, 4, 5, 3 viz Obrázek 49.



Obrázek 49: Čtvrtý krok prohlídky do hloubky

2.5.3 Průchod grafu do šířky

Průchod do šířky se označuje BFS z anglického Breadth First Search. [25]

Při průchodu grafem do šířky se nenavštívené vrcholy neukládají na zásobník, ale do fronty. Na rozdíl od průchodu do hloubky se následující vrchol nevybírámá souseď aktuálního vrcholu, ale vždy se vrchol vybírámá z fronty. Vrcholy jsou navštěvovány ve stejném pořadí, v jakém byly ukládány do fronty. [25]

Algoritmus průchodu grafu do šířky:

(1) Počáteční nastavení:

- *Fronta je prázdná.*
- *Aktuální vrchol v nastavíme na počáteční vrchol.*

(2) Navštívení aktuálního vrcholu:

- *Označíme aktuální vrchol jako navštívený.*
- *Postupně vložíme všechny nenavštívené sousední vrcholy aktuálního vrcholu do fronty.*

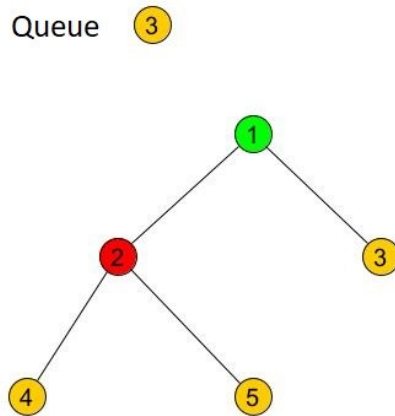
(3) Odebrání vrcholů z fronty:

- *Pokud je fronta prázdná, průchod je ukončen.*
- *Odebereme vrchol z fronty a ověříme, zda již nebyl navštívený, a pokud ano tak opět přejdeme na začátek kroku 3.*
- *Pokud odebraný vrchol ještě nebyl navštívený, nastavíme ho na aktuální vrchol a přejdeme na začátek kroku 2. [25]*

2.5.4 Příklad průchodu do šířky

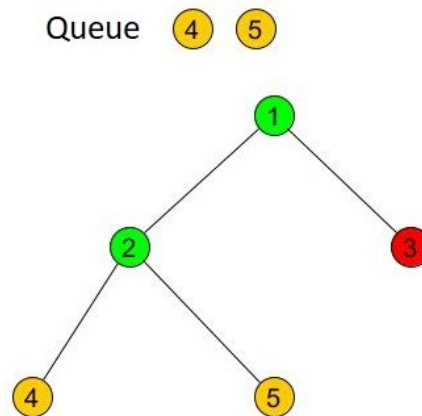
Průchod do šířky si ukážeme na grafu z předchozí ukázky průchodu do hloubky viz Obrázek 45.

Jako počáteční vrchol zvolíme vrchol 1 a označíme ho za navštívený. Vrchol 1 má dva nenavštívené sousedy, proto vrchol 2 a vrchol 3 uložíme do fronty a přejdeme ke kroku 3 odebereme vrchol 2 z fronty a nastavíme jako aktuální. Navštívené vrcholy jsou označeny zeleně a aktuální vrchol je označen červeně viz Obrázek 50.



Obrázek 50: První krok prohlídky do šířky

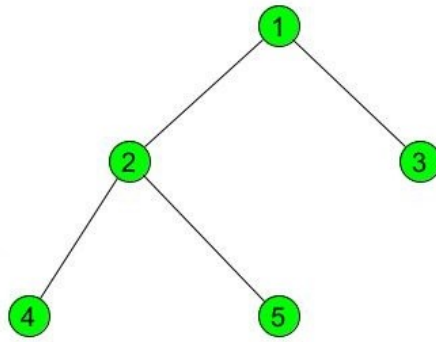
Aktuální vrchol, tedy vrchol 2, nastavíme jako navštívený. Vrchol 2 má opět dva nenavštívené sousedy vrcholy 4 a 5, vrcholy 4 a 5 tedy opět vložíme do fronty a přejdeme ke kroku 3 a to odebrání vrcholu 3 z fronty viz Obrázek 51.



Obrázek 51: Druhý krok prohlídky do šířky

Aktuální vrchol, tedy vrchol 3, nastavíme jako navštívený. Vrchol 3 nemá žádné nenavštívené sousedy, proto přejdeme ke kroku 3 a odebereme vrchol 4 z fronty, jelikož vrchol 4 nemá žádné nenavštívené sousedy, nastavíme ho jako navštívený a opět odebereme další vrchol z fronty, tedy vrchol 5. Vrchol 5 opět nemá žádné nenavštívené sousedy, a proto přejdeme ke kroku 3. Zjistíme, že fronta je prázdná, a proto je průchod u konce. Průchod tímto grafem do šířky je tedy posloupnost vrcholů 1, 2, 3, 4, 5 viz Obrázek 52.

Queue



Obrázek 52: Třetí krok průchodu do šířky

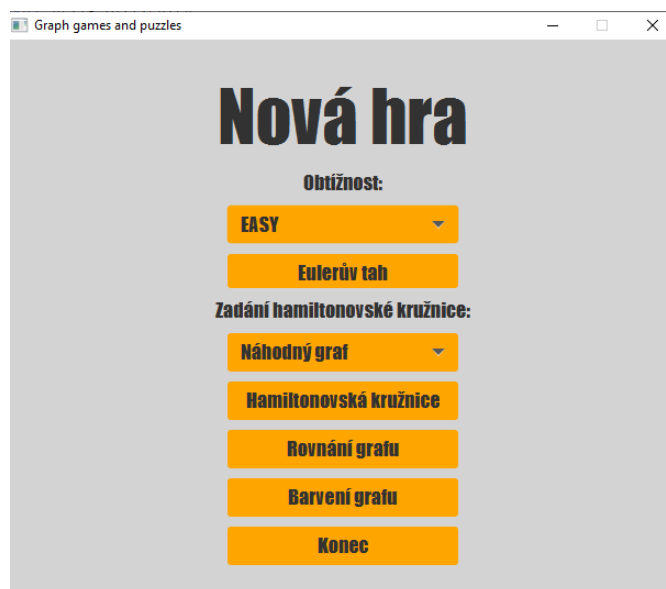
3 POPIS HERNÍ APLIKACE

3.1 Prerekvizity pro spuštění aplikace

Jedinou podmínkou pro spuštění aplikace je mít nainstalované JRE minimální verze 1.8. Aplikaci lze jednoduše spustit otevřením jar souboru *herniAplikace.jar* v příloženém zip archivu.

3.2 Popis aplikace

Po spuštění aplikace se zobrazí následující okno viz Obrázek 53.



Obrázek 53: Hlavní menu aplikace

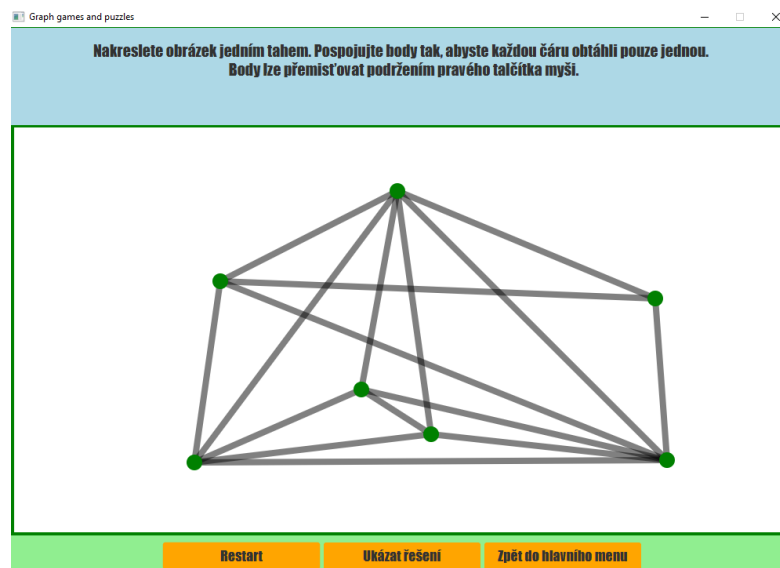
Hlavní menu obsahuje několik combo boxů a tlačítek:

- Combo box *obtížnost*, slouží k vybrání obtížnosti pro všechny typy her. Máme na výběr ze 3 obtížností a to *Easy*, *Medium* a *Hard*.
- Combo box *zadání hamiltonovské kružnice*, slouží k vybrání, zda se má graf generovat náhodně, anebo zda si chceme zahrát původní Hamiltonovu hru Icosian Game.
- Tlačítko *Eulerův tah* nám spustí hru, která spočívá v hledání Eulerova tahu.
- Tlačítko *Hamiltonovská kružnice*, nám spustí hru, která spočívá v hledání hamiltonovské kružnice.

- Tlačítko *Rovnění grafu*, nám spustí hru, která spočívá v hledání rovinného zobrazení grafu.
- Tlačítko *Barvení grafu*, nám spustí hru, která spočívá v obarvení grafu, tak aby sousední vrcholy nebyly obarveny stejnou barvou.
- Tlačítko *Konec*, které slouží k ukončení aplikace.

3.3 Eulerovský tah

Pro další ukázky budeme používat obtížnost *Easy*. Zvolíme tedy obtížnost *Easy* a klikneme na tlačítko *Eulerův tah* a otevře se nám následující okno.



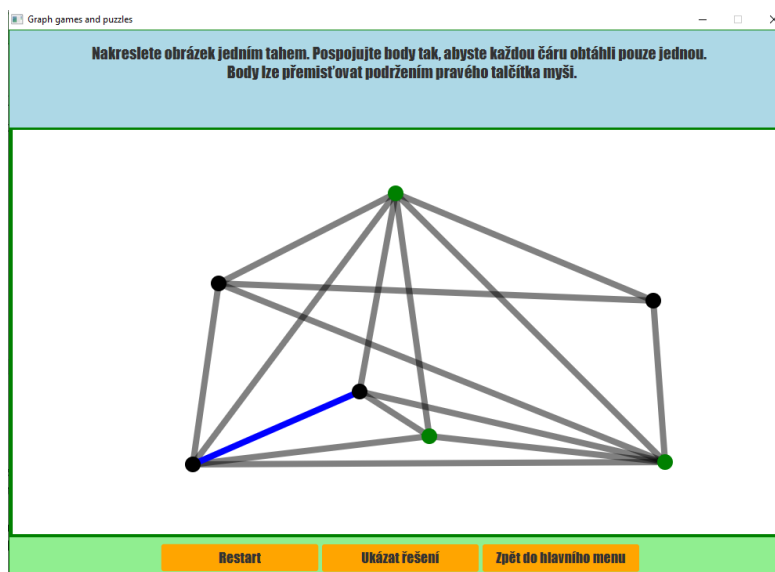
Obrázek 54: Okno Eulerův tah

V horní části okna můžeme vidět zadání, podle kterého má hráč hru dokončit. V dolní části vidíme tři tlačítka:

- Tlačítko *Restart*, pomocí kterého můžeme daný level restartovat.
- Tlačítko *Ukázat řešení*, pomocí kterého si může hráč, který si neví s daným levellem rady, nechat ukázat správné řešení. Po stisknutí tohoto tlačítka už mu ale nebude umožněno tento level dohrát.
- Tlačítko *Zpět do hlavního menu*, které slouží k návratu do hlavní nabídky.

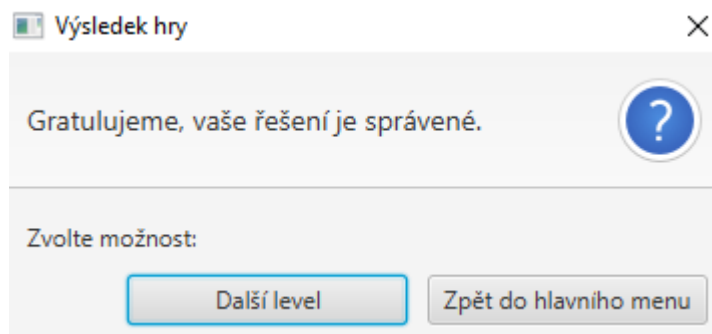
Ve střední části okna můžeme vidět vygenerovaný Eulerovský graf. Naším úkolem je pospojovat body tak, abychom prošli každou hranu právě jednou. Všechny body mají zelenou barvu, to znamená, že můžeme kliknout na jakýkoliv z nich. Po kliknutí na daný bod se nám

zobrazí sousední body námi zvoleného vrcholu, které jsou naše další možnosti. Po kliknutí na další zelený vrchol, se nám hrana mezi předchozím vrcholem a aktuálním vrcholem obtáhne viz *Obrázek 55*.



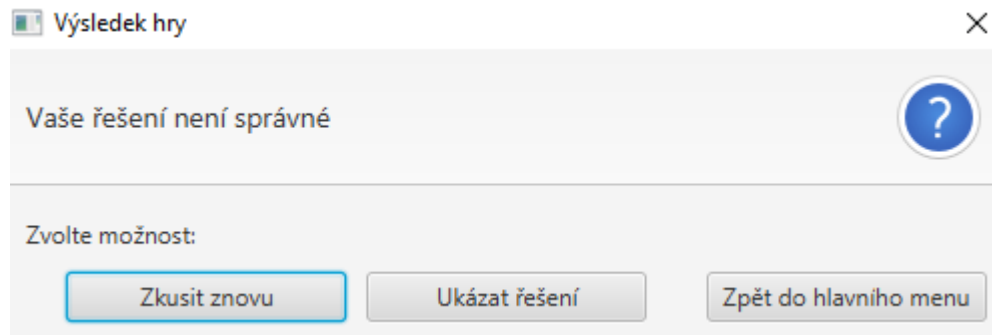
Obrázek 55:Spojování bodů

Pokud se nám podaří úspěšně dokončit Eulerův tah zobrazí se nám okno, které nám říká že naše řešení je správné. Máme na výběr dvě tlačítka, tlačítko *Další level*, které nám vygeneruje nové zadání se stejnou obtížností anebo tlačítko *Zpět do hlavního menu*, které nás vrátí zpět do hlavní nabídky viz *Obrázek 56*.



Obrázek 56: Správné řešení

Pokud se dostaneme do „slepé uličky“ zobrazí se nám okno, které nám říká, že naše řešení není správné. Máme na výběr tři tlačítka, tlačítko *Zkusit znovu*, které nám level restartuje a můžeme se pokusit znovu o správné řešení, tlačítko *Ukázat řešení*, které nám ukáže správné řešení, ale už se nemůžeme znovu pokusit o zvládnutí levelu, a tlačítko *Zpět do hlavního menu*, které nás přesune do hlavní nabídky.

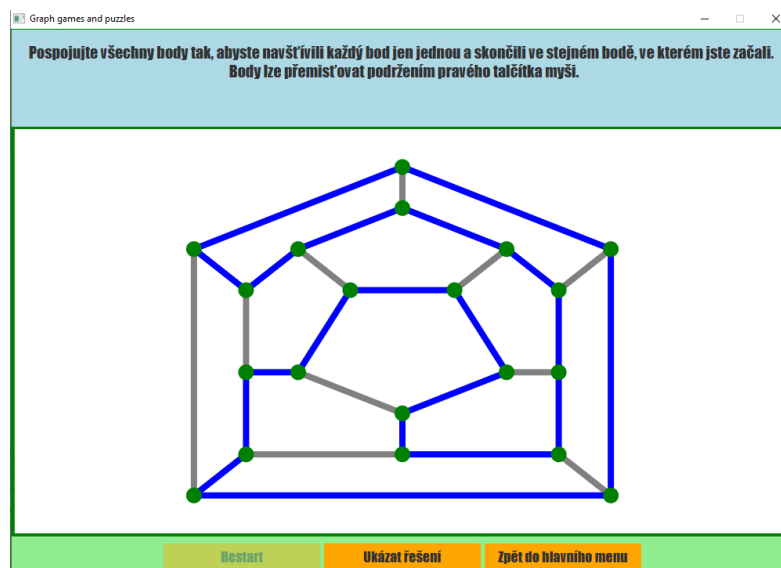


Obrázek 57: Špatné řešení

K vygenerování zadání pro hru Eulerovský tah je nejprve použit algoritmus BFS k ověření, zda je graf spojitý. Dále je kontrolována podmínka, zda graf obsahuje žádný nebo právě dva vrcholy lichého stupně, pokud ano je tento vygenerovaný graf použit jako zadání. K nalezení Eulerovského tahu, při kliknutí na tlačítko „Ukázat řešení“ je využit Fleuryho algoritmus. K ověření, zda hrana není mostem při provádění Fleuryho algoritmu a při kontrolování hráče je využit algoritmus DFS.

3.4 Hamiltonovská kružnice

Hra hledání hamiltonovské kružnice funguje na stejném principu jako Eulerův tah, pokud se dostaneme do slepé uličky hra končí, pokud se nám podaří najít hamiltonovskou kružnici vyhráli jsme. Příklad řešení hamiltonovské kružnice je například následující viz Obrázek 58.



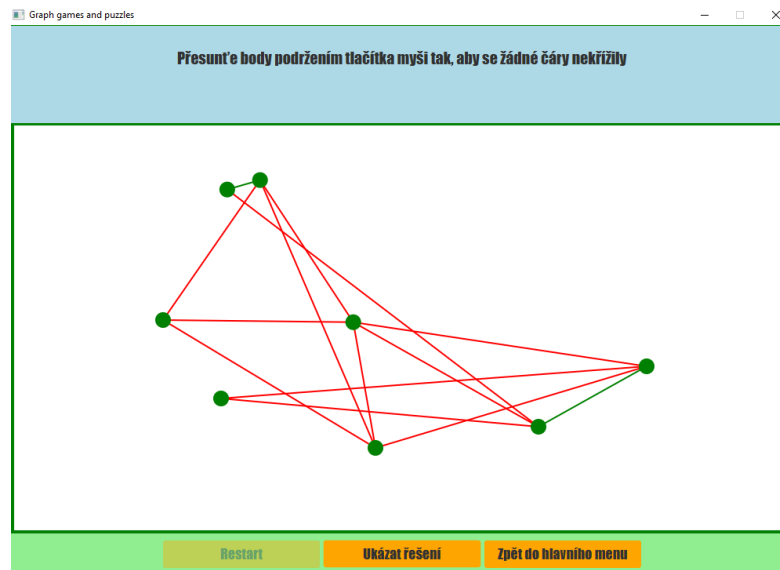
Obrázek 58: Okno hamiltonovská kružnice

K vygenerování zadání je opět použit algoritmus BFS k ověření spojitosti. Následně jsou kontrolovány nutné podmínky existence hamiltonovské kružnice. Pokud je tato podmínka

splněna, neznamená to, že je daný graf hamiltonovský, existence je ověřena naleznutím hamiltonovské kružnice pomocí backtracking algoritmu. Tato nalezená kružnice je následně ukázána po kliknutí na tlačítko „Ukázat řešení“.

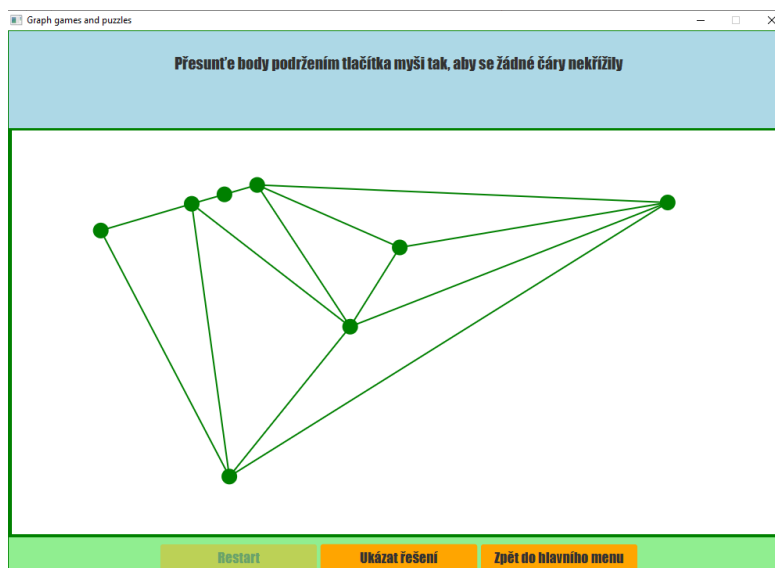
3.5 Hledání rovinného zobrazení

Další hra hledání rovinného grafu spočívá v tom, že musíme přesouvat vrcholy grafu držením tlačítka myši, dokud se žádné dvě hrany nekříží. Hrany, které se kříží jsou označeny červeně a hrany které se nekříží jsou označeny zeleně viz *Obrázek 59*.



Obrázek 59: Okno rovnání grafu

Řešení je správné, pokud jsou všechny hrany obarveny zeleně, to znamená že se žádné dvě hrany nekříží viz *Obrázek 60*.

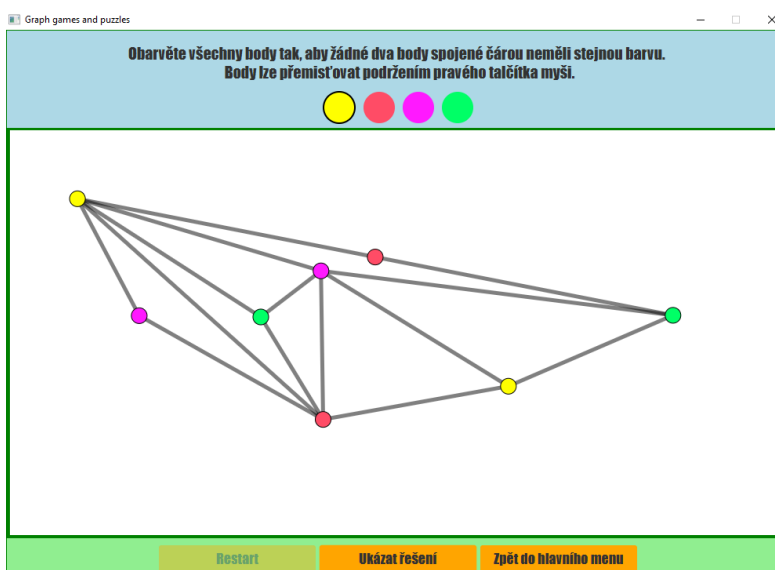


Obrázek 60: Okno řešení rovnání grafu

K vygenerování zadání je opět využit algoritmus BFS k ověření spojitosti grafu. Následně je vygenerován rovinný graf, tak aby se hrany nekřížily a následně je využita Kuratowského věta a je prováděna operace dělení hranou.

3.6 Barvení grafu

Poslední hra barvení grafu spočívá v barvení grafu barvami, které máme na výběr v horní části okna, tak aby sousední vrcholy nebyly obarveny stejnou barvou. Graf budeme barvit do té doby, než je graf obarven správně viz *Obrázek 61*.



Obrázek 61: Okno barvení grafu

K vygenerování zadání je opět použit algoritmus BFS k ověření spojitosti grafu následně je vygenerován rovinný graf, stejným způsobem jako v případě předchozím. K nalezení obarvení grafu po stisknutí tlačítka „Ukázat řešení“ je využit algoritmus sekvenčního barvení grafu.

4 TECHNICKÁ DOKUMENTACE

4.1 Použité technologie

4.1.1 Java

Herní aplikace byla vyvíjena v programovacím jazyce Java, aplikace byla vyvíjena a testována na verzi JRE 1.8, proto je doporučeno používat pro spuštění aplikaci právě verzi JRE 1.8.

4.1.2 JavaFX

Pro tvorbu uživatelského rozhraní (GUI) byla využita open source knihovna JavaFX.

4.1.3 Gluon Scene Builder

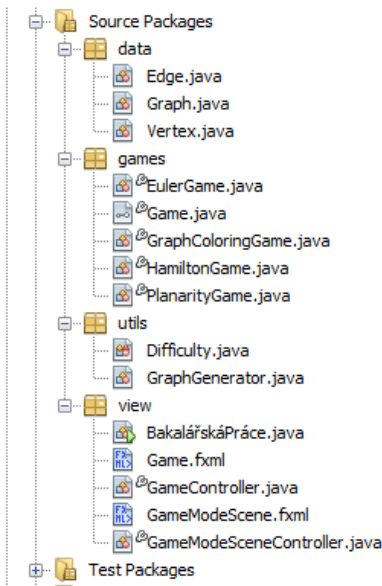
Ke grafickému designu JavaFX prvků byl použit open source nástroj Gluon Scene Builder verze 18.0.0.

4.1.4 NetBeans IDE 8.2

Jako vývojové prostředí pro psaní herní aplikace bylo použito NetBeans IDE 8.2.

4.2 Rozvržení aplikace

Aplikace je rozvržena do čtyř základních balíčků: *games*, *data*, *utils* a *view* viz Obrázek 62.



Obrázek 62: Balíčky

Balíček *games* obsahuje rozhraní *Game*, toto rozhraní implementují všechny ostatní algoritmy: *EulerGame*, *GraphColoringGame*, *HamiltonGame* a *PlanarityGame*. Balíček *data* obsahuje třídy, které slouží k reprezentaci grafu:

- Třída *Graph* slouží k reprezentaci grafu, obsahuje seznam hran a vrcholů.
- Třída *Vertex* slouží k reprezentaci vrcholu grafu, obsahuje označení vrcholu, informaci, zda byl vrchol navštíven, seznam hran nebo bod, kterým je vrchol reprezentován graficky
- Třída *Edge* slouží k reprezentaci hrany, obsahuje počáteční vrchol, koncový vrchol, informaci, zda byla hrana navštívena a čáru, kterou je hrana reprezentována graficky.

Balíček *utils* obsahuje pomocné třídy jako je Enum *Difficulty*, které slouží k reprezentaci obtížnosti pro generování grafu a třídu *GraphGenerator*, který generuje náhodné grafy jako zadání pro jednotlivé hry.

Balíček *view* obsahuje třídy, které slouží ke grafické reprezentaci herní aplikace, a to třídy *GameController* která slouží ke grafickému zobrazení dané hry a třídu *GameModeSceneController*, která slouží k reprezentaci hlavního menu.

4.3 Popis vybraných částí kódu

4.3.1 Generování náhodného grafu

```
public Graph generateGraph() {
    do {
        graph.getVertices().clear();
        graph.getEdges().clear();
        for (int i = 0; i < verticeNumber; i++) {
            graph.addVertice(new Vertex(i));
        }
        for (int i = 0; i < edgeNumber; i++) {
            Edge edge = generateEdge(graph.getVertices().size());
            graph.addEdge(edge);
            edge.getDestination().addEdge(edge);
            edge.getSource().addEdge(edge);
        }
    } while (!isContGraph());
    return graph;
}
```

Obrázek 63: Generovani náhodného grafu kód

4.3.2 Prohlídka do šířky

Metoda *generateGraph* generuje náhodný graf podle parametrů *verticeNumber* a *edgeNumber* dokud není graf spojitý, k ověření spojitosti je využita prohlídka grafu do šířky.

```
private List<Vertex> breadthFirst(Vertex v) {
    List<Vertex> visitedVertices = new ArrayList<>();
    v.setVisited(true);
    LinkedList<Vertex> queue = new LinkedList<>();
    queue.offer(v);
    while (!queue.isEmpty()) {
        Vertex vertice = queue.poll();
        visitedVertices.add(vertice);
        for (Vertex neighbor : graph.getNeighbours(vertice)) {
            if (!neighbor.isVisited()) {
                neighbor.setVisited(true);
                queue.offer(neighbor);
            }
        }
    }
    return visitedVertices;
}
```

Obrázek 64: Prohlídka grafu do šířky kód

Metoda *breadthFirst* projde daný graf do šířky a vrátí seznam navštívených vrcholů.

4.3.3 Hledání Eulerova tahu

```
private void findPath(Vertex start) {  
    for (Edge edge : start.getEdges()) {  
        if (!edge.isVisited()) {  
            Vertex nextVertex = graph.getNeighbour(start, edge);  
            if (isEdgeValid(start, edge)) {  
                edge.setVisited(true);  
                eulerianPath.add(nextVertex);  
                findPath(nextVertex);  
            }  
        }  
    }  
}
```

Obrázek 65: Hledání Eulerova tahu kód

Metoda *findPath* postupně prochází všechny hrany vstupního vrcholu a pokud hrana nebyla navštívena a zároveň platí, že odebrání hrany nezpůsobí rozpad grafu na dvě komponenty je koncový vrchol hrany přidán do Eulerova tahu.

4.3.4 Hledání hamiltonovské kružnice

```
private boolean findCycle(int index) {  
    if (index == graph.getVertices().size()) {  
        return graph.getNeighbours(  
            HamiltonCycle.get(index - 1)).contains(  
                graph.getVertex(0));  
    }  
    for (int i = 1; i < graph.getVertices().size(); i++) {  
        if (isValidVertex(graph.getVertex(i))) {  
            HamiltonCycle.add(graph.getVertex(i));  
            if (findCycle(index + 1)) {  
                return true;  
            }  
            HamiltonCycle.remove(index);  
        }  
    }  
    return false;  
}
```

Obrázek 66: Hledání hamiltonovské kružnice kód

Metoda *findCycle* přidá do cesty validní vrchol, a následně rekurzivně zavolá funkci *findCycle*. Pokud je nalezen validní vrchol, přidá se do cesty a je vráceno true, pokud ne je vráceno false a pomocí „backtrackování“ je vrchol odebrán z cesty a je prozkoumána další možnost.

4.3.5 Barvení grafu

```
private void colorGraph() {  
    Vertex current;  
    while (!isEveryVerticeColored()) {  
        current = getHighestDegreeVertice();  
        current.setVisited(true);  
        current.getCircle().setFill(findPossibleColor(current));  
        current.getCircle().setStrokeWidth(1);  
    }  
}
```

Obrázek 67: Barvení grafu kód

Metoda *colorGraph* barví vrcholy, dokud nejsou všechny vrcholy obarveny. Pomocná metoda *getHighestDegreeVertice* vrátí vrchol s nejvyšším stupněm. Metoda *findPossibleColor* najde takovou barvu, kterou lze vrchol obarvit tak, aby neměl stejnou barvu jako jeho sousední vrcholy.

ZÁVĚR

Hlavním důvodem psaní této práce bylo vytvoření herní aplikace, která je založena na teorii grafů. Tento cíl byl dle mého názoru splněn, aplikace obsahuje čtyři hry, a to Kreslení jedním tahem, hledání Hamiltonovské kružnice, hledání rovinného zobrazení grafu a barvení rovinného grafu. Aplikace umožňuje hráči zvolit jedni ze tří obtížností pro každou hru a to Easy, Medium a Hard, podle které se generují různě složité grafy. Hráč si během hraní vyzkouší vyřešit různé problémy teorie grafů herní a zábavnou formou. I když se hry zdají být jednoduché opak je pravdou, například při zvolení obtížnosti Hard u hry rovnání grafu může být někdy nalezení rovinného zobrazení zábavou na několik dlouhých minut. Sám jsem při testování aplikace i přes znalost všech algoritmů měl někdy s obtížností Hard problémy, proto jsem ji nakonec trochu zjednodušil. Když jsem dal aplikaci otestovat některým příslušníkům rodiny či kamarádům, kteří nemají žádné znalosti z teorie grafů, dokázali se u hry zabavit na poměrně dlouhou dobu, takže si myslím že herní aplikace splnila svůj účel.

Herní aplikaci bych vytknul vzhled uživatelského rozhraní, který by se dal v budoucnu vylepšit, aby byla herní zkušenost ještě lepší. Dále by se dala aplikace jednoduše rozšířit o další hry jednoduchým vytvořením nové třídy, která implementuje rozhraní reprezentující algoritmus. Dle mého názoru jsou tyto hry velmi zábavné a zároveň užitečné, protože si uživatel například u Eulerovského tahu může zobrazit řešení krok po kroku a obohatit tak své zkušenosti teorie grafů touto zábavnou formou. Z tohoto důvodu si myslím, že práce splnila svůj účel a může být přínosem pro širokou veřejnost.

POUŽITÁ LITERATURA

- [1] BALAKRISHNAN, V.K. *Schaum's Outline of Theory and Problems of Graph Theory* [online]. 1997 edition. New York: McGraw-Hill Education, 1997 [cit. 2023-04-14]. ISBN 0-07-005489-4. Dostupné z: <https://kashanu.ac.ir/Files/GraphTheoryBalakrishnan1.pdf>
- [2] KOVÁŘ, Petr. *Úvod do Teorie grafů* [online]. Ostrava, 2012 [cit. 2023-04-14]. Dostupné z: https://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/uvod_do_teorie_grafu.pdf. Příprava přednášek a cvičení. Vysoká škola báňská – Technická univerzita Ostrava a Západočeská univerzita v Plzni.
- [3] MATOUŠEK, Jiří. *Kapitoly z diskrétní matematiky*. Vyd. 1. Praha: Karolinum, 2002, [cit. 2023-04-14]. ISBN 80-246-0084-6. Dostupné z: https://lms.umb.sk/plugin-file.php/58452/mod_lesson/page_contents/9220/Nesetril.pdf
- [4] VON BELL, Matias. *Highlights from the History of Graph Theory* [online]. Helsinki, 2015 [cit. 2023-04-14]. Dostupné z: https://wiki.helsinki.fi/download/attachments/154152712/Matias_Von_Bell_Graph_Theory.pdf?version=1&modification-Date=1455564456184&api=v2. Výukový materiál. Universiti of Helsinki.
- [5] MAŠTEROVÁ, Ludmila. *Teorie grafů*. Brno, 2016. Bakalářská práce. Masarykova univerzita, Přírodovědecká fakulta, Vedoucí práce RNDr. Pavel Šišma, Dr.
- [6] *Teorie grafů* [online]. Ostrava: Vysoká škola Báňská, 21.6.2006 [cit. 2023-04-14]. Dostupné z: <http://books.fs.vsb.cz/SystAnal/texty/21.htm>
- [7] KÁRNÁ, Lucie. *Teorie grafů* [online]. Praha, 2018 [cit. 2023-04-14]. Dostupné z: <https://zolotarev.fd.cvut.cz/y1tg/ctrl.php?act=show,file,26358>. Výukový materiál. České vysoké učení technické v Praze, fakulta Dopravní.
- [8] JIROVSKÝ, Lukáš. *Teorie grafů* [online]. Praha, 2010 [cit. 2023-04-14]. Dostupné z: <https://teorie-grafu.cz/>. Diplomová práce. Univerzita Karlova, Matematicko-fyzikální fakulta. Vedoucí práce RNDr. Pavla Pavlíková, Ph.D.
- [9] DVOŘÁK, Zdeněk. *Sledy, tahy, cesty. Stupně a skóre* [online]. Praha, 2018 [cit. 2023-04-14]. Dostupné z: <https://iuuk.mff.cuni.cz/~rakdver/dm/lesson8.pdf>. Výukový materiál. Univerzita Karlova, Matematicko-fyzikální fakulta.

- [10] MÁŠILKO, Lukáš. *Diskrétní matematika I* [online]. Brno, 2017 [cit. 2023-04-14]. Dostupné z: https://is.muni.cz/el/ped/podzim2018/MA2BP_PDM1/um/02_Souvislost.pdf. Výukový materiál. Masarykova Univerzita.
- [11] ŠÁMAL, Robert. *Rovinné grafy* [online]. Praha, 2018 [cit. 2023-04-14]. Dostupné z: <https://prase.cz/library/RovinneGrafyRS/RovinneGrafyRS.pdf>. Výukový materiál. Univerzita Karlova.
- [12] PRATHIK, A., K. UMA a J. ANURADHA. An Overview of application of Graph theory. In: *International Journal of ChemTech Research* [online]. CODEN (USA): ChemTech, 2016, s. 247 [cit. 2023-04-16]. ISSN 0974-4290. Dostupné z: https://www.researchgate.net/publication/301920836_An_overview_of_application_of_graph_theory
- [13] PAOLETTI, Teo. Leonard Euler's Solution to the Königsberg Bridge Problem. *Convergence* [online]. The College of New Jersey: Mathematical Association of America, May 2011 [cit. 2023-04-16]. Dostupné z: <https://www.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem>
- [14] Seven Bridges of Königsberg. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2023-04-17]. Dostupné z: https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg?oldid=130498025
- [15] Eulerian Cycles: Why Are They So Unique, and Are They Significant to Us in the 21st Century?. *Towards Data Science* [online]. Indie: Jaival Patel, 2019 [cit. 2023-04-26]. Dostupné z: <https://towardsdatascience.com/eulerian-cycles-why-are-they-so-unique-and-are-they-any-significant-to-us-in-the-21st-century-3ca489af585c>
- [16] ZELENKOVÁ, Jana. *Efektivní výpočty vlastností neorientovaných grafů*. Praha, 2021. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická. Vedoucí práce RNDr. Marko Genyk-Berezovskyj.
- [17] DEMLOVÁ, Marie. *Logika a grafy* [online]. Praha, 2018 [cit. 2023-04-14]. Dostupné z: <https://math.fel.cvut.cz/en/people/demlova/lgr/p-lgr816.pdf>. Výukový materiál. České vysoké učení technické v Praze, Fakulta elektrotechnická.

- [18] KRÁTKÝ, Radim. *Hamiltonovské grafy* [online]. Hradec Králové, 2020 [cit. 2023-04-14]. Dostupné z: <https://theses.cz/id/wy704j/STAG92896.pdf>. Diplomová práce. Univerzita Hradec Králové, Fakulta informatiky a managementu. Vedoucí práce RNDr. PhDr. Antonín Slabý, CSc.
- [19] KOVÁŘ, Petr. *Teorie grafů* [online]. Ostrava, 2023 [cit. 2023-04-14]. Dostupné z: http://homel.vsb.cz/~kov16/files/skriptum_teorie_grafu_rozsirene.pdf. Výukový materiál. Vysoká škola báňská – Technická univerzita Ostrava a Západočeská univerzita v Plzni.
- [20] The Icosian Game. *The Puzzle Museum* [online]. [cit. 2023-04-23]. Dostupné z: <https://www.puzzlemuseum.com/month/picm02/200207icosian.htm>
- [21] Šachové pojmy. *Chess.com* [online]. c2023 [cit. 2023-04-23]. Dostupné z: <https://www.chess.com/terms/knights-tour-chess>
- [22] KOVÁŘOVÁ, Alena. *PROBLÉM ČTYŘ BAREV* [online]. Plzeň, 2013 [cit. 2023-04-14]. Dostupné z: https://dspace5.zcu.cz/bitstream/11025/7132/1/Problem_ctyr_barev.pdf. Diplomová práce. Západočeská univerzita v Plzni, Fakulta pedagogická. Vedoucí práce PhDr. Šárka Pěchoučková, Ph.D.
- [23] HERMAN, Vladimír. *Barvení grafů* [online]. Hradec Králové, 2016 [cit. 2023-04-14]. Dostupné z: <https://theses.cz/id/m05si6/18620014>. Diplomová práce. Univerzita Hradec Králové, Fakulta informatiky a managementu. Vedoucí práce RNDr. Andrea Ševčíková.
- [24] *Algoritmy a programovací techniky* [online]. Praha: Prometheus, 1995 [cit. 2023-04-24]. ISBN 80-858-4983-6. Dostupné z: http://mff.cz/data/ADS_PTAPT.pdf
- [25] VEČERKA, Arnošt. *GRAFY A GRAFOVÉ ALGORITMY* [online]. Olomouc, 2007 [cit. 2023-04-24]. Dostupné z: https://phoenix.inf.upol.cz/esf/ucebni/Grafy_a_grafove_algoritmy.pdf. Výukový materiál. Univerzita Palackého, Přírodovědecká fakulta.
- [26] KAMINANI, Sridher. *FINDING HAMILTONIAN CYCLES* [online]. Bowling Green, Kentucky, 2005 [cit. 2023-04-14]. Dostupné z: <https://core.ac.uk/download/pdf/43618678.pdf>. Diplomová práce. Western Kentucky University, The Faculty of the Department of Computer Science. Vedoucí práce Dr. Uta Ziegler.
- [27] RAK, Josef. *Teorie grafů*. Univerzita Pardubice, c2011-2020.

- [28] PŘIBYLÍK, Jakub. Vrcholová barvení grafů [online]. Ostrava, 2017 [cit. 2023-05-09]. Dostupné z: <https://core.ac.uk/download/pdf/94772315.pdf>. Bakalářská práce. Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky. Vedoucí práce Mgr. Tereza Kovářová, Ph.D.