



Dokumentace k projektu pro předmět

Grafové algoritmy

Demonstrační úloha - Fleuryho algoritmus

11. prosince 2014

Autor: Daniel Javorský, Michal Ruprich

Kontakt: {xjavor14,xrupri00}@stud.fit.vutbr.cz

Fakulta Informačních Technologií

Vysoké Učení Technické v Brně

Obsah

1	Úvod	2
2	Vymezení problému	2
2.1	Popis klíčových pojmů	2
2.1.1	Eulerovské tahy	2
2.1.2	Slovní popis Fleuryho algoritmu	2
2.1.3	Fleuryho algoritmus	2
2.1.4	Kdy je hrana most	2
2.2	Řešení hledání mostu	3
3	Návrh a implementace aplikace	3
3.1	Popis uživatelského rozhraní	3
3.2	Ovládání průchodu grafem	4
3.3	Vizualizace průchodu grafem	4
4	Závěr	4

1 Úvod

Tento dokument popisuje návrh a implementaci aplikace, demonstrující využití Fleuryho algoritmu při hledání eulerovských tahů v souvislých neorientovaných grafech. Aplikace umožňuje uživateli načítat předem vytvořené grafy uložené ve formátu xml nebo graphml, navíc je možné v aplikaci přímo vytvářet vlastní grafy. Načtené či vytvořené grafy je pak možné po krocích procházet dopředu i nazpět a sledovat hledání eulerovského tahu.

2 Vymezení problému

Tato kapitola se bude zabývat klíčovými slovy a odbornými názvy, které se používají v této dokumentaci.

2.1 Popis klíčových pojmů

Stručný popis pojmů, které jsou klíčové pro pochopení fungování algoritmu. Některé pojmy jsou zároveň důležité pro pochopení fungování aplikace.

2.1.1 Eulerovské tahy

Eulerovský tah je tah, který prochází všemi hranami právě jednou. Graf, který obsahuje eulerovský tah nazýváme eulerovský.

Theorem 2.1.1. *Neorientovaný graf G má eulerovský tah, právě když je G souvislý a počet vrcholů s lichým stupněm je 0 nebo 2. [2]*

2.1.2 Slovní popis Fleuryho algoritmu

Mějme graf G , který je eulerovský, tedy splňuje podmínku z teoremu 2.1.1. Pokud má graf 2 vrcholy s lichým stupněm, začne algoritmus v jednom z těchto vrcholů a skončí v druhém vrcholu s lichým stupněm. V případě, že má graf 0 vrcholů s lichým stupněm, vybere se náhodný uzel, ve kterém algoritmus začne a v tomto vrcholu také skončí.

Algoritmus hledá eulerovský tah grafem. V každém vrcholu ve kterém se momentálně nachází musí rozhodnout, jakou hranou se má vydat. Pokud z vrcholu vede pouze jedna hrana, vybere se právě ta. Pokud z vrcholu vede více hran, musí se vybrat taková hrana, po jejímž odebrání zůstane prohledávaný podgraf souvislý.

2.1.3 Fleuryho algoritmus

Theorem 2.1.2. *Pokud G je souvislý graf splňující teorem 2.1.1, pak sled W , který je výstupem Fleuryho algoritmu, je eulerovský tah grafu G .*

2.1.4 Kdy je hrana most

V algoritmu 1 je zmíněn pojem most. Mějme souvislý graf G , F zatím neprozkoumaný souvislý podgraf grafu G a hranu e . Pokud se po odebrání hrany e z grafu F stane nesouvislý graf, pak je e most. V průběhu Fleuryho algoritmu je tedy nutné určit, zda je zkoumaná hrana most.

Algoritmus 1: FLEURYHO ALGORITMUS

Input: Spojitý graf G a počáteční uzel u

Output: Eulerovský tah W grafu G začínající (případně i končící) v u

- 1: polož $W := u, x := u, F := G$
 - 2: **while** $z\ x$ vychází nějaké hrany **do**
 - 3: vyber hranu $e := xy$, kde xy je hranou vycházející z x a zároveň e je jediná hrana vycházející z x nebo e není most v F
 - 4: přidej y do W , odeber hranu e z F , polož $x:=y$
 - 5: **end while**
 - 6: **return** W
-

2.2 Řešení hledání mostu

Zda je hrana most se zjišťuje následovně. Pomocí Depth First Search (dále jen DFS) se zjistí počet vrcholů zatím neprozkoumaného podgrafu, které jsou dostupné z právě prohledávaného vrcholu. Z podgrafu se odebere zkoumaná hrana a pomocí DFS se opět zjistí počet dostupných vrcholů z prohledávaného vrcholu. Pokud je počet vrcholů po odebrání hrany menší, než před odebráním, znamená to, že hrana je most a nesmí tedy být vybrána pro odebrání z grafu.

3 Návrh a implementace aplikace

Demonstrační aplikace je napsána v jazyce Java. Pro reprezentaci grafů je použita knihovna jgraphx.

3.1 Popis uživatelského rozhraní

Uživatelské rozhraní je rozděleno do tří hlavních částí. Na pravé straně je hlavní ovládací panel, ve spodní části okna se nachází ovládací prvky pro průchod grafem a uprostřed je pracovní plocha pro vytváření a editaci grafů.

Na hlavním panelu se nachází soubor tlačítek pro ovládání celé aplikace. V horní části hlavního panelu je oblast, kam se vypisují informace o prováděných operacích a informace o vytvářeném grafu. Ve spodní části je oblast, ve které je při průchodu grafem vidět náhled na původní graf. Následuje krátký popis hlavních ovládacích prvků:

- **New graph** – vytvoření nového okna pro editaci grafu
- **Save** – uložení vytvářeného grafu ve formátu xml
- **Load** – načtení předvytvořeného grafu uloženého ve formátu xml nebo graphml
- **Erase graph** – vymazání vytvářeného grafu
- **Save as image** – uložení grafu ve formě obrázku s příponou png
- **Help** – zobrazí nápovědu
- **Start** – umožní uživateli průchod grafem jako demonstraci algoritmu

3.2 Ovládání průchodu grafem

Stisk tlačítka start v hlavní nabídce uživateli umožní přístup k ovládacím prvkům ve spodní části aplikace, za předpokladu, že vytvořený graf splňuje podmínky uvedené v teorému 2.1.1. Ovládací prvky ve spodní části jsou určeny k postupnému procházení vytvořeným grafem. Zde je krátký popis jejich funkcí:

- **Step back** – provede se jeden krok zpět při procházení grafem. Tlačítko není aktivní, pokud je průchod na začátku
- **Start over** – navrátí se na začátek průchodu grafem a nahradí procházený graf původním
- **Step forward** – provede se jeden krok vpřed při procházení grafem. Pokud se algoritmus nachází v posledním procházeném vrcholu grafu, opětovné stisknutí tlačítka zobrazí výslednou posloupnost vrcholů eulerovského tahu. Tlačítko je neaktivní, pokud je průchod na konci
- **Re-edit** – aktivují se hlavní ovládací prvky, uživateli je umožněno v grafu provádět další změny

3.3 Vizualizace průchodu grafem

Průchod grafem je naznačen pomocí postupného odebrání hran z prohledávaného grafu. Zelená barva označuje právě prohledávaný vrchol. Ve spodní části hlavního panelu je možné během průchodu vidět původní podobu grafu. Během průchodu se ukládá posloupnost navštívených vrcholů která je na konci vypsána v horní části hlavního panelu.

4 Závěr

Cílem projektu bylo vytvořit aplikaci demonstrující chod zadaného algoritmu a různé operace nad grafem. Všechny tyto možnosti výše popsána aplikace splňuje.

Reference

- [1] Bondy, J.A.; Murty, U.S.R.: *Graph Theory*, Springer, 2008, ISBN 978-1-84628-969-9
- [2] Křivka, Zbyněk; Masopust, Tomáš: *Grafové algoritmy*
<https://www.fit.vutbr.cz/study/courses/GAL/public/gal-handouts.pdf>