



IMPLEMENTACE ALGORITMU PRO NALEZENÍ NEJMENŠÍ
KOSTRY GRAFU

TÝM XKULIN01

Obsah

1 Úvod.....	2
2 Zadání	2
3 Návrh a implementace	2
3.1 Postup algoritmu	2
3.2 Složitost implementovaného algoritmu	2
4 Datové struktury	3
4.1 Seznam	3
4.2 Množina	3
4.3 Uzel	3
4.4 Hrana	3
5 Členění implementačního řešení.....	3
6 Práce v týmu.....	3
7 Reprezentace grafu	4
8 Spouštění aplikace	4

1 Úvod

Jméno souboru: IAL_kostra_grafu

Identifikace projektu: IAL, Náhradní projekt - 07. Minimální kostra grafu

Členové týmu:

Kulinkovich Andrei (xkulin01)

Marochkina Elena (xmaroc00)

Tréšek Roman (xtrese00)

Hierarchie souborů:

- src - hlavní adresář

- implementace algoritmu rozdělena do několika jednotlivých souborů, pro jednotlivé části grafu

- graphs - adresář s testovými soubory

Datum vytvoření: 12.11.2023, datum poslední změny: 27.11.2023

2 Zadání

07 . Minimální kostra grafu - Vytvořte program pro hledání kostry grafu s minimálním ohodnocením pro ohodnocené neorientované grafy.

3 Návrh a implementace

K řešení jsme využili Primova algoritmu, jeho teoretická složitost je $O(E + V \cdot \log_2 E)$, kde E je počet hran a V je počet vrcholů v grafu.

3.1 Postup algoritmu

Ze začátku se náhodně vybere jeden uzel jako počáteční, zbylé uzly se zapíší do množiny nezpracovaných uzlů. Všechny hrany, které vedou z počátečního uzlu se zapíší na seznam dostupných hran. Algoritmus projde seznam dostupných hran a vybere tu s nejnižším ohodnocením, která ale musí vést do ještě nezpracovaného uzlu. Hrana je poté odstraněna ze seznamu dostupných hran a uzel se odstraní z množiny nezpracovaných uzlů. Zpracovávaná hrana se přidává do fronty hran reprezentujících postupně rostoucí kostru grafu. Uzel do kterého zpracovávaná hrana vedla se odstraní z množiny nezpracovaných uzlů. Všechny hrany dostupné z nového uzlu jsou zapsány na seznam dostupných hran. Algoritmus probíhá tak dlouho, dokud nezpracuje všechny uzly.

3.2 Složitost implementovaného algoritmu

Složitost námi implementovaného algoritmu je $O(V^2 + E)$, protože jsme využili implementaci pomocí seznamu sousedů oproti rychlejší variantě s využitím haldy.

Implementace pomocí seznamu sousedů je pomalejší, protože prochází seznam uzlů a hledá jeho minimum.

4 Datové struktury

4.1 Seznam

Implementovali jsme obousměrně vázaný seznam. Pro potřebu implementace seznamu dostupných hran, který je potřeba k realizaci algoritmu. Seznam vznikl na základě toho, jak byl prezentován v přednáškách.

4.2 Množina

Vytvořili jsme jednoduchou množinu pro podporu implementace uzlů. Do množiny se ukládají hodnoty zpracovávaných uzlů a také zpracované uzly. Množina je zde vhodná, protože hodnoty se v těchto případech nemohou opakovat.

4.3 Uzel

Tato struktura je také vlastní strukturou. V této struktuře jsou uloženy údaje o hranách, které vedou do nebo z tohoto uzlu, dále je zde uložen počet těchto hran a také jednotlivé názvy uzlů. Pole takových uzlů tvoří graf.

4.4 Hrana

Tato struktura je vlastní, tj. vyvinuli jsme ji speciálně pro tento projekt. Účelem této struktury je interpretovat hrany mezi uzly. Tato struktura má tři prvky: vzdálenost mezi dvěma uzly, uzel, z něhož hrana vychází a uzel, do něhož vede spojnice. Rád bych poznamenal, že předpokládáme, že mezi dvěma uzly jsou dvě hrany (mají stejnou vzdálenost, ale v hranách se střídají pouze samotné uzly).

5 Členění implementačního řešení

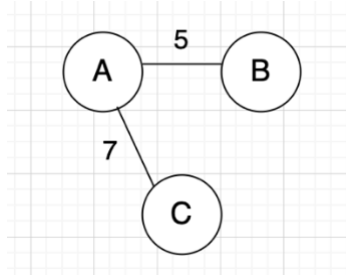
Algoritmus	<code>algorithm. {c, h}</code>
Hrana	<code>edge. {c, h}</code>
Seznam	<code>list. {c, h}</code>
Uzel	<code>node. {c, h}</code>
Čtečka	<code>reader. {c, h}</code>
Množina	<code>set. {c, h}</code>
Funkce main	<code>main. c</code>

6 Práce v týmu

Člen týmu	Přidělená práce
Andrei Kulinkovich	Implementace algoritmu, vytvoření struktur, dokumentace
Elena Marochkina	Implementace algoritmu, testování
Roman Tréšek	Implementace algoritmu, dokumentace, prezentace

7 Reprezentace grafu

V testových souborech jsou grafy reprezentovány pomocí jejich hran. Následující graf by byl reprezentován jako (A,B,5),(A,C,7).



8 Spouštění aplikace

Překlad se provádí pomocí příkazu `make`.

Pro testy máme bash skript `test.sh`, který se stane spustitelný po příkazu **`chmod +x test.sh`**, poté stačí pro spuštění testu napsat **`./test.sh`**.

Grafy jsou brány z podadresáře **`graphs`**, nebo lze při překladu napsat jako argument textový soubor s grafem.