IAL

# Dokumentácia k náhradnému projektu

### Číslo a názov projektu :

### 7. Minimální kostra grafu

### Zadanie :

* *Kostra grafu* je podgraf, který je stromem a který obsahuje všechny vrcholy původního grafu. Z této definice vyplývají následující tvrzení. Kostra grafu je souvislý graf. Kostra grafu má o jednu hranu méně než má vrcholů. Nesouvislý graf nemá kostru. Kostra grafu neobsahuje kružnici.
* Vytvořte program pro hledání kostry grafu s minimálním ohodnocením pro ohodnocené neorientované grafy.
* Pokud existuje více řešení, nalezněte všechna. Výsledky prezentujte vhodným způsobem. Součástí projektu bude načítání grafů ze souboru a vhodné testovací grafy. V dokumentaci uveďte teoretickou složitost úlohy a porovnejte ji s experimentálními výsledky.

## Obsah :

* Výber vhodného algoritmu
* Teoretická zložitosť vybraného algoritmu
* Porovnanie výsledkov

### Autori :

1. xpospi95 - Michal Pospíšil
2. xtimko01 - Nikola Timková
3. xbalaz08 - František Balázsy

## Výber vhodného algoritmu :

Pre riešenie zadanej problematiky existuje rada algoritmov, našou úlohou bolo vybrať najvhodnejší pre pokrytie všetkých požiadaviek vyplývajúcich zo zadania s prihliadnutím na jeho technické parametre a jednoduchosť implementácie. Pre prehľad uvádzame krátku rešerš najznámejších algoritmov :

### Borůvkov algoritmus

* Pažravý algoritmus vynájdený pri riešení problematiky konštrukcie elektrickej siete na Morave. Značne obmedzujúci, pretože predpokladá grafy s hranami kladných a rôznych hodnôt. Môže však byť paralelizovateľný, čo mu dáva výhodu oproti ostatným v prípade dostatku výpočtových jednotiek. Tvorí základ probabilistického algoritmu bežiaceho v lineárnom čase, ktorý však funguje len pre grafy s neizolovanými vrchoľmi.

### Primsov algoritmus

* Pažravý algoritmus pre neorientované grafy vynájdený Vojtechom Jarníkom. Svoje najväčšie využitie nachádza v prípadoch veľmi hustých grafov s veľa vrcholmi, kde beží v O(E + V log V). Táto časová zložitosť však platí len v prípade, že spracováva vstupné dáta v podobe Fibonacciho haldy a zoznamu susednosti. Pokiaľ by sme algoritmus spustili nad maticou susednosti by v najhoršom prípade dosiahol časovú zložitosť O(|V|2).

### Kruskalov algoritmus

* Pažravý algoritmus vhodný na typické, redšie grafy. Vyznačuje sa veľkou implementačnou výhodou oproti Primsovmu algoritmu pretože dokáže spracovať priamo maticu susednosti. Taktiež má nižšiu časovú zložitosť nad bežnými grafmi, a to O(E \* log V).

Pre riešenie nášho zadania sme si vybrali Kruskalov algoritmus, pretože vyhovuje povahe našich vstupných dát, vyznačuje sa najjednoduchšou implementovateľnosťou a dosahuje najlepšiu časovú zložitosť nad bežnými grafmi. Taktiež je možné pomocou neho jednoducho nájsť viacero riešení.

**Teoretická zložitosť Kruskalovho algoritmu :**