

Programovací strategie: Síť

Tomáš Maršálek

30. října 2011

1 Zadání

Jakožto velký expert na všechny druhy algoritmů a řešení obtížných problémů (firma Ferda Mravenec, práce všeho druhu, s.r.o.) jste byli pověřeni firmou vlastnící počítačovou sítí kontrolou optimality informačních toků po síti. Počítače jsou již zapojeny do sítě, je známa cena přenosu v každém komunikačním kanálu (může jít o kapacitu, délku, cenu spojení...). Síť zahrnuje až 1000 uzlů, ceny jsou kladná čísla. Propojení v nejhorším případě může tvořit úplný graf. Zajímá nás minimalizace ceny při posílání informace – musíte najít, které kanály z existujících se mají používat a které naopak doporučujete nepoužívat.

Váš úkol založený na tomto příběhu je následující:

Implementujte algoritmus založený na greedy strategii a na tom, co víte o řešení problému MST, určete, jestli jde o přesné nebo přibližné řešení, určete jeho složitost. Není požadována žádná vizualizace sítě, stačí prosté načtení vstupu z text. souboru ve tvaru:

nV nE (tj. 1. řádek udává počet uzlů a počet hran)

e_i e_j w_{ij} (tj. n řádek se zadáním hran a jejich ohodnocením – předpokládá se, že ohodnocení je stejné v obou směrech)

Výstup na obrazovku nebo opět do souboru (výstupem je počet hran doporučených k používání a jejich výčet v nějakém přijatelném formátu). Získáte 6 bodů. Pokud vaše řešení bude nejvýše $O(E \log E)$, kde E je počet kanálů, získáte další 4 body.

2 Řešení

Úloha je řešena pomocí Primova algoritmu pro MST. Implementace zjistí výsledek v $O(E \log E)$. Z prioritní fronty se odebere nejvýše $2E$ hran (datová struktura grafu obsahuje hrany oběma směry) a odebrání jedné hrany stojí $O(\log E)$, protože prioritní fronta obsahuje nejvýše $2E$ hran a je implementována binární haldou. Výstup algoritmu je přesné řešení.

3 Implementace

Program je v Java SE 6. Pokud není graf souvislý, neexistuje žádná kostra grafu a program takovou situaci odhalí. Pokud v grafu existují hrany se stejným ohodnocením, bude existovat více možných řešení, program vypíše pouze jednu z nich. Výstupní formát je stejný jako vstupní. Testování proběhlo na několika malých testovacích souborech úspěšně.

3.1 Příklad použití

```
java -jar mst.jar vstupni.graf -o vystupni.graf
```

Reference

- [1] *Prof. Erik Demaine, Prof. Charles Leiserson*
Introduction to Algorithms (SMA 5503), Lecture 16: Greedy Algorithms, Minimum Spanning Trees (video) 2005
<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-introduction-to-algorithms-sma-5503-fall-2005/video-lectures/lecture-16-greedy-algorithms-minimum-spanning-trees/>