

Cvičení #5

Miloš Chromý

chromy@ktiml.mff.cuni.cz

1. **Nejtěžší kamion.** Chcete dopravit zásobu roušek do centrálního skladu. Máte k dispozici kamion s libovolnou kapacitou, ale silnice bohužel nejsou stavěny na neomezenou váhu. Každá silnice má svoji orientaci a hodnotu, která udává maximální váhu auta, které může po silnici projet. Najděte cestu z výrobní do centrálního skladu, po které projede nejtěžší dodávka se zásobami. *Napište si pseudokód relaxačního algoritmu.*
2. **Dijkstra.**
 - (a) Najděte orientovaný ohodnocený graf s právě jednou zápornou hranou a bez záporného cyklu, na kterém Dijkstra selže. (V závislosti na implementaci buď otevře vrchol znovu nebo nevypočte nejkratší cestu správně.)
 - (b) Najděte graf s celočíselnými vahami na kterém Dijkstra poběží exponenciálně dlouho.
 - (c) **Dů.** Mějme orientovaný graf $G = (V, E, c)$ s kapacitami $c : E \rightarrow \{1, \dots, L\}$. Navrhněte datovou strukturu založenou na příhrádkách pro Dijkstraův algoritmus takovou, že algoritmus bude mít složitost $O(nL + m)$ na grafu G . Pokuste se i o to, abyste potřebovali pomocnou paměť $O(n + m + L)$. (Podrozdělení hran není správné řešení). Složitost dokažte.
3. **Záporák.** Pomocí Belman-Fordova algoritmu detekujte libovolný záporný cyklus dosažitelný z daného vrcholu v v grafu G a vypište ho.
4. **Letadla.** Chceme se dostat z konference v Číně z domů. Pro každý let místo a čas odletu a místo a čas přiletu. V čase t jsme na letišti ve Wu-chanu a chceme se dostat do Prahy.
 - (a) Jak se dostat domů co nejrychleji?
 - (b) Nechce se nám trávit moc času na přestupech, aby nás nechali ještě odletět. Jak se dostat domů co nejrychleji pomocí co nejméně letadel?
5. **Pravděpodobnost.** Každé spojení má pravděpodobnost úspěchu přenosu $0 \leq p_i \leq 1$. Pravděpodobnost úspěchu pro cestu v_1, \dots, v_k je $\prod (v_i, v_{i+1})$. Chcete najít cestu, po které je nejvíce pravděpodobné, že uspějete. Jak ji najdete?
6. **Vycházky za časů korony.** Ve vašem městě je zákaz vycházení. Lidé mohou pouze na nejkratší procházky okolo svého bydliště. Navrhněte algoritmus, jak pro každého obyvatele nalézt jeho povolenou procházku. Mapa města je orientovaný graf. Výstup by měl být pro každý vrchol délka nejkratšího cyklu na kterém se daný vrchol nachází.