

ADS2 Jiří Kalvoda: Cvičení 5

Informace k cv. jsou na <https://kam.mff.cuni.cz/~jirikalvoda/vyuka/25z/ads2>.

Toky v sítích

Ford-Fulkersonův algoritmus:

- 1) Najdi (libovolnou) cestu zdroj–stok z nenasyčených hran
- 2) Spočítej, o kolik nejvýše lze na této cestě zvýšit tok
- 3) Zvyš tok
- 4) Opakuj, dokud nějaká cesta existuje

Hrana je **nenasyčená**, pokud tok je menší než kapacita (může být způsobeno i tím, že něco teče v protisměru).

U úloh nezabývajících se Fordovým-Fulkersonovým algoritmem můžete předpokládat, že existuje algoritmus, který pro zadanou síť najde maximální tok v polynomiálním čase vzhledem k počtu vrcholů a hran.

Dinicův algoritmus:

- 1) Začneme s prázdným tokem f .
- 2) Iterativně vylepšujeme tok pomocí zlepšujících toků: (vnější cyklus)
 - 1) Sestrojíme síť rezerv: vrcholy a hrany jsou tytéž, kapacity jsou určeny rezervami v původní síti. Dále budeme pracovat s ní.
 - 2) Najdeme nejkratší st -cestu. Když žádná neexistuje, skončíme.
 - 3) Pročistíme síť, tj. ponecháme v ní pouze vrcholy a hrany na nejkratších st -cestách.
 - 4) Najdeme v pročištěné síti blokující tok f_B :
 - 1) $f_B \leftarrow$ prázdný tok
 - 2) Postupně přidáváme st -cesty: (vnitřní c.) $\mathcal{O}(mn)$
 - 1) Najdeme st -cestu. Např. hladově – „rovnou za nosem“.
 - 2) Pošleme co nejvíce po nalezené cestě.
 - 3) Smažeme nasycené hrany.
 - 4) Dočistíme síť.
 - 5) Zlepšíme f podle f_B

Úloha 1 (Implementace): Ukažte, jak v Dinicově algoritmu naprogramovat hledání blokujícího toku a dodatečného čištění sítě rezerv jako (opakované) prohledávání do hloubky nebo do šířky.

Úloha 2 (Lepší odhad): Dokažte, že máme-li v síti celočíselné kapacity omezené konstantou, Dinicův algoritmus doběhne v čase $\mathcal{O}(nm)$.

Úloha 3 (Poslanci): V parlamentu s n poslanci je m různých klubů. Jeden poslanec může být členem mnoha různých klubů. Každý klub nyní potřebuje zvolit svého předsedu a tajemníka tak, aby všichni předsedové a tajemníci byli navzájem různé osoby (tedy aby nikdo „neseděl na více křeslech“). Navrhněte algoritmus, který zvolí všechny předsedy a tajemníky, případně oznámí, že řešení neexistuje.

Mimochodem, za jakých podmínek je existence řešení garantována?

Úloha 4 (Hranová souvislost): Graf je hranově k -souvislý, jestliže je souvislý a po odebrání libovolných $k - 1$ hran bude stále souvislý. Najděte algoritmus, který zjistí pro jaké největší k je daný graf hranově k -souvislý. Může se vám hodit hledání toků v až $\mathcal{O}(n)$ sítích.

Úloha 5 (Továrny): Uvažujme továrny $T_1, \dots, T_p$ a obchody $O_1, \dots, O_q$. Všechny továrny vyrábějí stejný druh zboží. Továrna T_i za den vyrobí t_i kusů, obchod O_j spotřebuje o_j kusů. Navíc známe bipartitní graf určující, která továrna může dodávat zboží kterému obchodu. Najděte efektivní algoritmus, který zjistí, zda je požadavky obchodů možné splnit, aniž by se překročily výrobní kapacity továren, a pokud je to možné, vypíše, ze které továrny se má přepravit kolik zboží do kterého obchodu.

Úloha 6 (Figurky): Je dána šachovnice $k \times k$, kde některá políčka jsou nepřístupná. Celý dolní řádek je obsazen figurkami, které se mohou hýbat o jedno pole dopředu, šikmo vlevo dopředu, či šikmo vpravo dopředu. V jednom tahu se všechny figurky naráz pohnou (mohou i zůstat stát na místě), na jednom políčku se však musí vyskytovat nejvýše jedna figurka. Ocitne-li se figurka na některém políčku horního řádku šachovnice, zmizí. Navrhněte algoritmus, který najde minimální počet tahů takový, že z šachovnice dokážeme odstranit všechny figurky, případně oznámí, že řešení neexistuje.

Můžete nejprve určit horní odhad počtu tahů.

Úloha 7 (Nekonečno): Najděte síť s reálnými kapacitami, na níž Fordův-Fulkersonův algoritmus nedoběhne. Lze dokonce zařídit, aby k maximálnímu toku ani nekonvergoval.