5. CVIČENÍ Z ADS2

Toky v sítích – Dinic

Příklad první Dokažte, že pro jednotkové kapacity Dinicův algoritmus doběhne v čase $O(nm)$.

Příklad druhý Dokažte následující *užitečné lemma:* Pokud Dinicův algoritmus pustíte na celočíselných kapacitách a počátečním tokem velikosti A, a on nalezne tok velikosti B, tak algoritmus běžel čas $O((B - A) \cdot n + mn)$.

Příklad třetí Dokažte, že pro celočíselné kapacity omezené konstantou doběhne Dinicův algoritmus v čase $O(mn)$, čili vlastně v čase $O(f(C) \cdot mn)$.

Příklad čtvrtý Řekněme, že nám $f(C)$ pro celočíselné kapacity nestačí, a chtěli bychom variantu Dinice, která doběhne opravdu rychle v závislosti na kapacitách.

Hezkým trikem je *škálování kapacit.* Představíme si kapacity v binárním zápisu stejné délky ($11 =$ `1011`, $3 =$ `0011`), a pak budeme tok konstruovat iterativně takto: nejprve najdeme nula-jedničkový tok pro kapacity rovny přesně nejvyššímu bitu, pak nalezený tok zdvojíme, kapacity také zdvojíme a přičteme jedničky pro kapacity, která mají na druhé nejvyšší pozici jedničku.

Pro 11 tedy první kapacita bude `1`, potom `10`, v třetím kroku `101` a nakonec `1011`.

Pojďme ho zanalyzovat!

1. Nejprve dokažte, že pokud na začátku iterace je tok f_Z a na konci je f_K, tak $|f_K - f_Z| \leq O(m)$.
2. Jakou tedy máme složitost škálovacího algoritmu?

Příklad pátý Sestrojte vrstevnatou síť, v níž hledání blokujícího toku trvá $\Omega(nm)$.

Příklad šestý Sestrojte síť, na níž Dinicův algoritmus provede $\Omega(n)$ fází.

Příklad sedmý Zkombinujte předchozí dvě cvičení a vytvořte síť, na níž Dinicův algoritmus běží v čase $\Omega(n^2 m)$.

Příklad osmý Existuje mnoho problémů, které jdou řešit hledáním maximálního toku; asi jste už třeba slyšeli o maximálním párování v bipartitním grafu.

Jeden z nejtěžších problémů, který jde převést na toky, je tenhle, jehož řešení si nikdy nepamatuji:

„*Máme doly $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ a továrny $\mathcal{T} = \{T_1, \ldots, T_m\}$. Otevření jednoho dolu nás stojí d_i korun, a každá továrna nám vydělá t_j korun. Nicméně každá továrna j vyžaduje podmnožinu dolů $P_j \subseteq \mathcal{D}$, a vydělá oněch t_j korun jen tehdy, pokud jsme všechny její doly otevřeli. (Jeden důl stačí otevřít jen jednou, pak je použitelný pro všechny továrny, které ho potřebují.)*

Našim úkolem je spočítat, které doly otevřít, abychom vydělali co nejvíce peněz. “

Pomozte mi připomenout si, jak vyřešit tento problém pomocí hledání toků v sítích!

Hint: Jediné, co si pamatuji, je že je chytřejší dívat se na minimální řez a rozmyslet si, proč mi spočítá, co chci.

Info na konec: Za 14 dní bude písemka!