

Kombinatorika a Grafy 1 - Cvičení 7

Jan Soukup

13.11.-15.11.2023

<https://kam.mff.cuni.cz/~soukup/vyuka/2324/KAG/>

1 Toky v sítích

Síť je uspořádaná čtveřice (G, z, s, c) , kde $G = (V, E)$ je orientovaný graf, neboli $E \subseteq V \times V$, z a s jsou dva různé vrcholy grafu G (zvané *zdroj* a *stok*) a *kapacita* $c: E \rightarrow \mathbb{R}_0^+$ je funkce ohodnocující hrany. *Tok v síti* je každá funkce $f: E \rightarrow \mathbb{R}$ splňující $0 \leq f(e) \leq c(e)$ pro každou hranu $e \in E$ a splňující

$$\sum_{v:(u,v) \in E} f(u, v) = \sum_{v:(v,u) \in E} f(v, u)$$

pro každý vrchol $u \in V$ mimo stok a zdroj. *Velikost toku* je

$$w(f) = \sum_{v:(z,v) \in E} f(z, v) - \sum_{v:(v,z) \in E} f(v, z).$$

Řezem nazveme podmnožinu E , po jejímž odstranění neexistuje orientovaná cesta ze zdroje do stoku. *Kapacita řezu* R je $c(R) = \sum_{e \in R} c(e)$.

Jako *rezervu hrany* e na nějaké cestě P ze z do s označíme $r(e) = c(e) - f(e)$ pro hranu e orientovanou po směru P a $r(e) = f(e)$ pro hranu orientovanou proti směru P . Cesta P je pak *zlepšující*, pokud všechny její hrany mají kladnou rezervu.

Algorithm 1.1: FORD–FULKERSON(G)

$f \leftarrow$ nulový tok

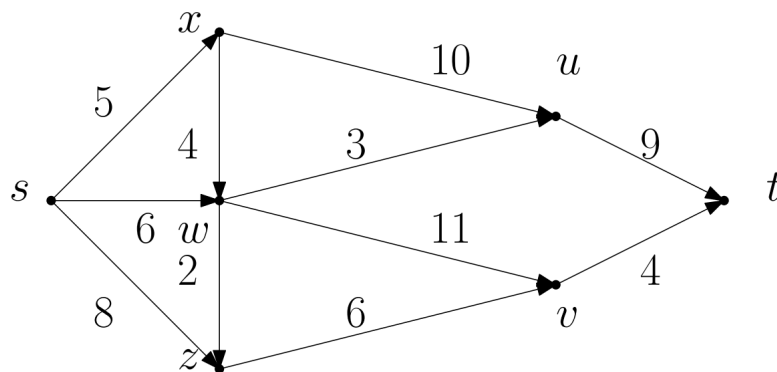
while existuje zlepšující cesta P ze z do s

do $\left\{ \begin{array}{l} \epsilon_P \leftarrow \min_{e \in E(P)} r(e) \\ \text{Zvětšíme tok } f \text{ podél } P \text{ o } \epsilon_P \text{ (každé hraně } e \text{ po směru zvětšíme} \\ f(e) \text{ a hranám proti směru zmenšíme } f(e)). \end{array} \right.$

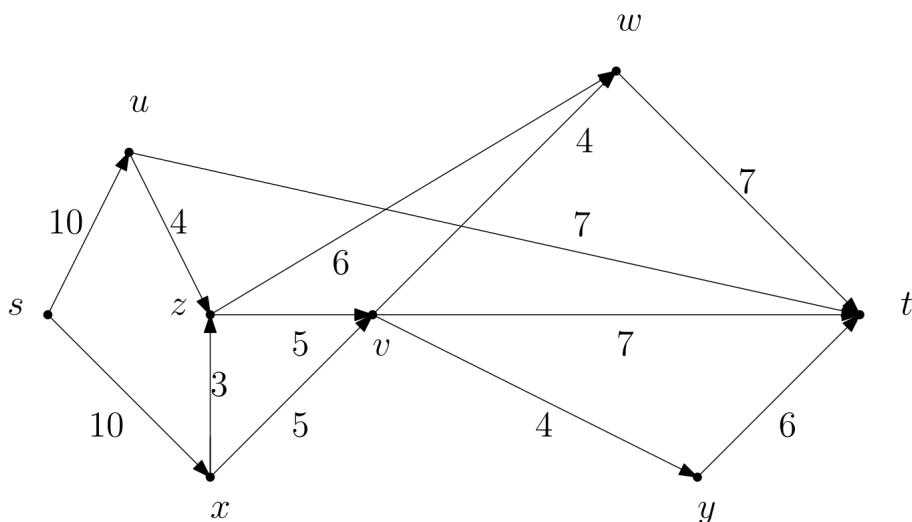
Vrať tok f .

Příklad 1. Najděte tok maximální velikosti v následujících sítích (Fordovým–Fulkersonovým algoritmem nebo uhádněte) ze zdroje s do stoku t . Nalezněte také řezy minimální kapacity a ověřte tak, že dané toky mají skutečně maximální velikost.

(1)



(2)



Příklad 2.

- (a) Najděte síť (a posloupnost konkrétních použitých zlepšujících cest), na které F.-F. algoritmus nemusí nedospět ke správnému výsledku, pokud mu povolíme používat jen orientované zlepšující cesty.
- (b) Najděte posloupnost sítí (a posloupnosti použitých zlepšujících cest), na které má F.-F. algoritmus exponenciální časovou složitost (vzhledem k počtu bitů potřebných k uložení grafu a kapacit).
- (c*) Najděte síť (a posloupnost konkrétních použitých zlepšujících cest), na které F.-F. algoritmus neskončí a ani nekonverguje k maximálnímu toku.