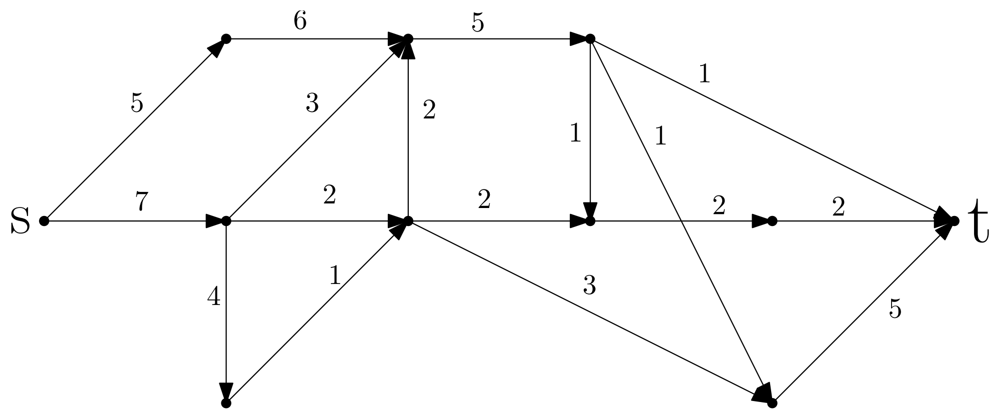
Příklad první Nalezněte maximální s,t-tok a minimální s,t-řez pro následující ohodnocený orientovaný graf:



Příklad druhý Najděte příklad sítě s nejvýše 10 vrcholy a 10 hranami, na níž Ford-Fulkerson provede více než milion iterací.

Příklad třetí Profesor Forderson nemá rád síť rezerv a chce navrhnout algoritmus, který pracuje jen se zlepšujícími cestami v původní síti. Pokusil se chybný algoritmus (který zlepšuje tok pouze po směru hran) zachránit tím, že bude vybírat nejkratší zlepšující cesty. Ukažte, že zlepšovák pana profesora nefunguje!

Příklad čtvrtý Jak už znáte z přednášky, toky v sítích mají spoustu zajímavých vlastností jako dualita s řezy, certifikát optimality, polynomiální algoritmus pro celočíselná řešení a další.

Kuriozitou je, že tyto vlastnosti úzce souvisí s tím, že problém toku v sítích se dá namodelovat jako maximalizace soustavy lineárních nerovnic s reálnými proměnnými. Zkusíme si to tedy namodelovat.

Co to znamená namodelovat? Představte si to tak, že budeme mít na vstupu jeden konkrétní orientovaný graf s kapacitami, a hledáme algoritmus, který graf zpracuje a vypíše na výstup soustavu lineárních nerovnic.

Tyto nerovnice budou mít nějaké proměnné – v našem případě bude reálná proměnná x_{uv} pro každou orientovanou hranu uv, a nerovnice samozřejmě budou mít i nějaké konstanty a nějaké pravé strany – to budou čísla, které algoritmus nějak odvodí z grafu (například kapacity atd.).

Naše úkoly jsou:

1. Popište soustavu lineárních nerovnic, která bude perfektně popisovat celý prostor maximálních i nemaximálních toků. Co to znamená: pro vstupní graf postavte soustavu lineárních nerovnic tak, že *každé řešení* soustavy odpovídá nějakému toku.
2. Jakmile máme soustavu výše, zkuste k ní vymyslet jednu *lineární funkci* se stejnými proměnnými takovou, že kdybyste našli hodnotu, které maximalizují tuto funkci, tak jste našli maximální tok.

Příklad pátý Algoritmus Ford-Fulkerson obvykle nemá dobrou časovou složitost. Umíme o ni alespoň něco říci?

1. Připomeňte si argument, že v síti s celočíselnými kapacitami se algoritmus vždy zastaví.
2. Dokažte, že na síti, kde jsou všechny kapacity 1, doběhne v rychlém čase (jakém?);

Příklad šestý Přímočará implementace algoritmu Ford-Fulkerson bude nejspíš graf prohledávat do šířky, takže vždy najde nejkratší nenasycenou cestu. Pak překvapivě platí, že algoritmus zlepší tok jen $O(nm)$-krát, než se zastaví. Pojďme to dokázat!

Návod k důkazu: Nechť $l(u)$ je vzdálenost ze zdroje do vrcholu u po nenasycených hranách. Nejprve si rozmyslete, že $l(u)$ během výpočtu nikdy neklesá. Pak dokažte, že mezi dvěma nasyceními libovolné hrany uv se musí $l(u)$ zvýšit. Proto každou hranu nasytíme nejvýše $O(n)$-krát.

Poznámka na závěr: jaká bude výsledná složitost algoritmu FF?