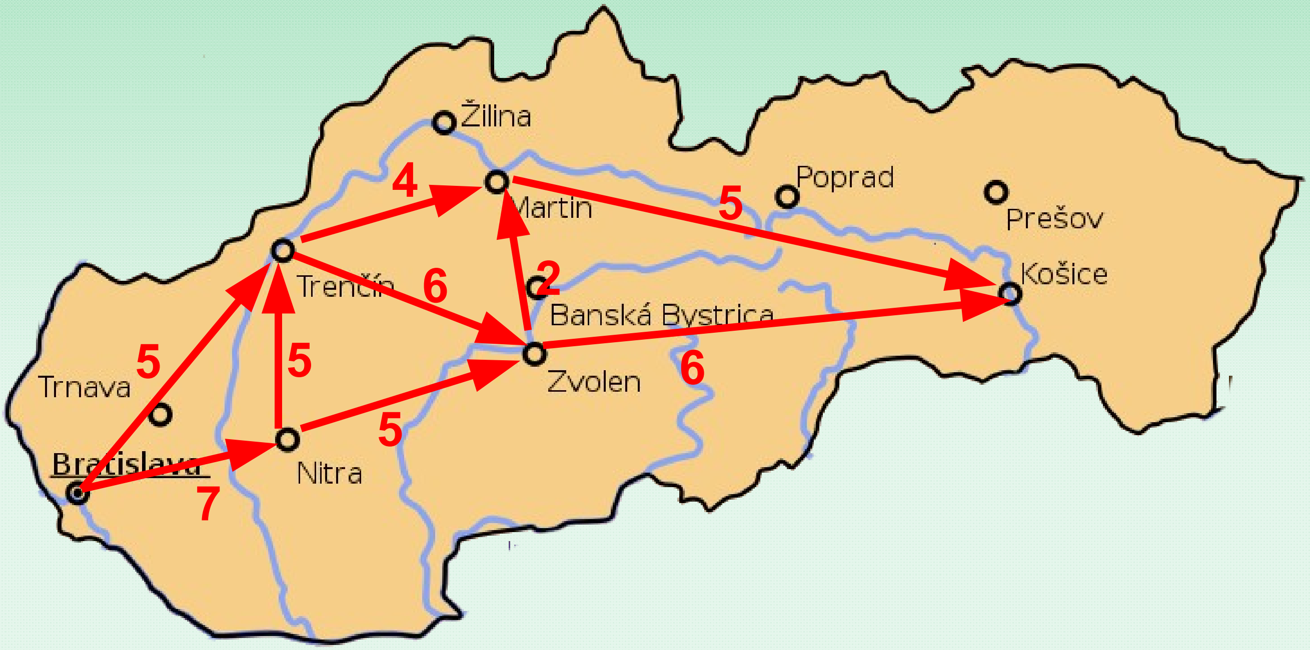
O čom dnes?

- Maximálny tok a minimálny rez
- Párovanie (Matching)

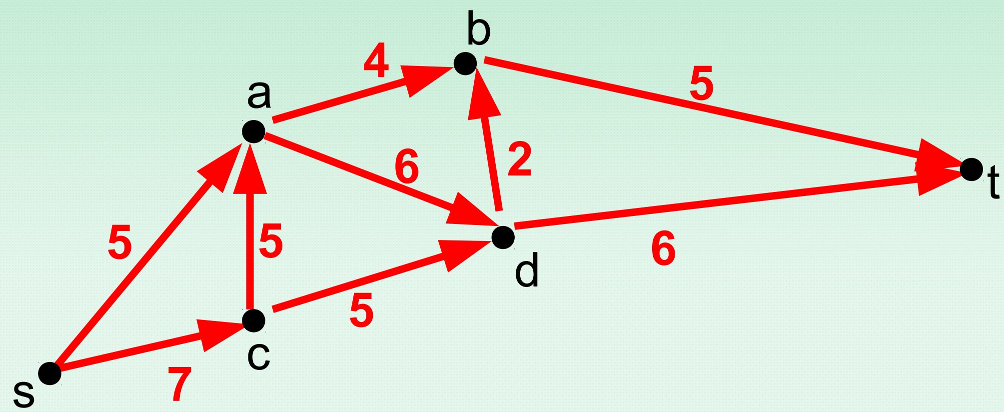
Maximálny tok

Maximálny tok



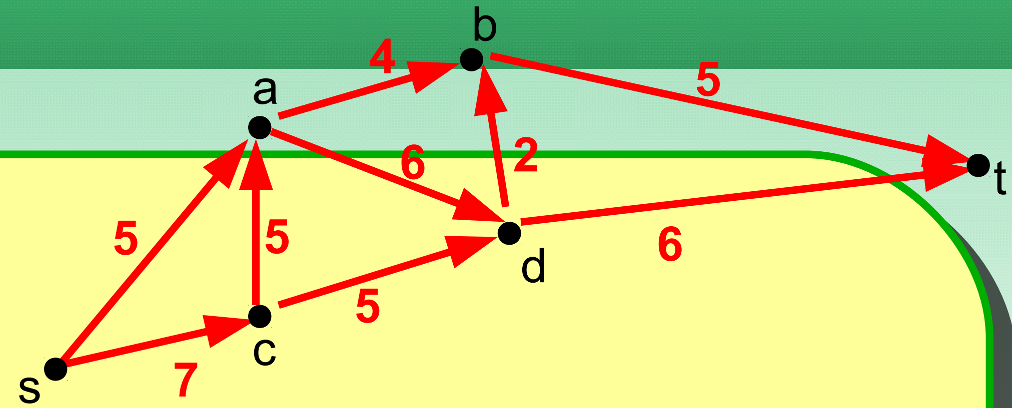
Koľko tovaru dokážem prepraviť z Bratislavy do Košíc pri uvedených transportných kapacitách?

Maximálny tok



Aký je maximálny tok z bodu s do bodu t pri uvedených kapacitách jednotlivých hrán?

Transportná sieť



Nech $N=(V,E)$ je orientovaný spojitý graf bez slučiek, potom N nazývame **transportná sieť**, ak v N

a) existuje práve jeden vrchol $s \in V$, ktorý nemá vstupné hrany ($inDeg(s) = 0$) a nazývame ho **zdroj** (source).

b) existuje práve jeden vrchol $t \in V$, ktorý nemá vstupné hrany ($outDeg(t) = 0$) a nazývame ho **spotrebič** (sink).

c) hrany v sieti majú priradenú nezápornú celočíselnú váhu, ktorú nazývame kapacita ($c(e) = c(v,w)$).

Tok

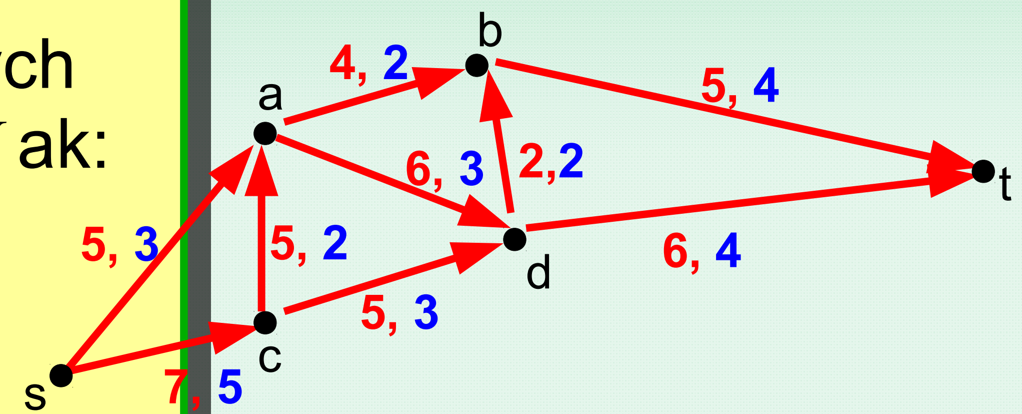
Nech $N=(V,E)$ je transportná sieť, funkcia f z E do nezáporných celých číselnazýva **tok** pre sieť N ak:

a) $\forall\, e \in E\ f(e) \leq c(e)$

b) $\forall\, v \in V \setminus \{s,t\}$

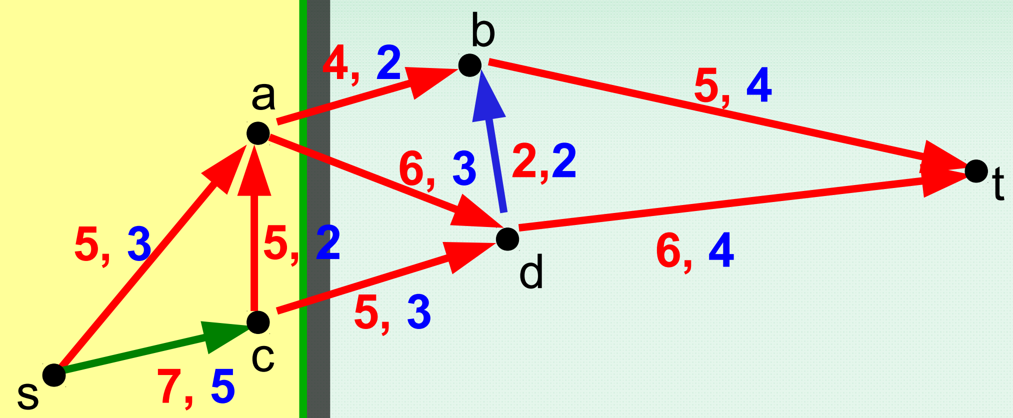
$$\sum_{w \in V} f(w,v) = \sum_{w \in V} f(v,w)$$

ak hrana v,w neexistuje $f(v,w) = 0$



Tok

Hranu e v sieti nazývame nasýtenou, ak $f(e) = c(e)$.
Ak $f(e) < c(e)$, ide o nenasýtenú hranu



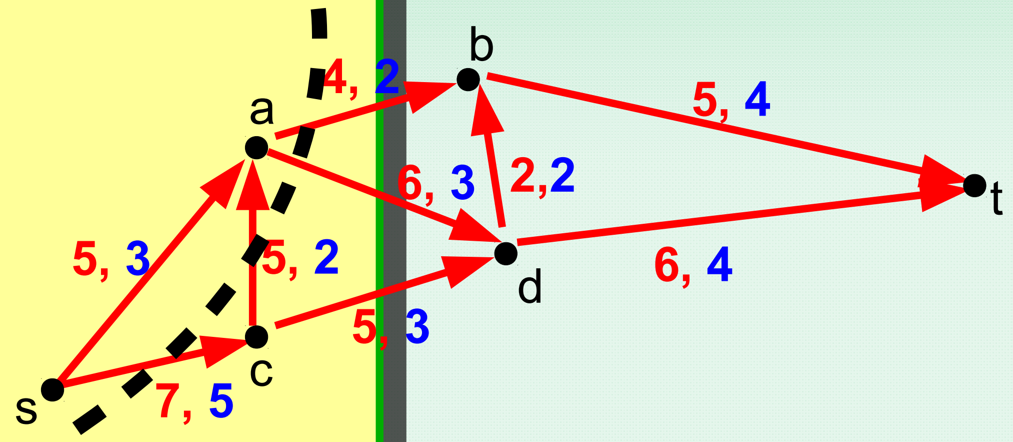
Hodnotu toku vypočítame ako

$$val(f) = \sum_{v \in V} f(s, v)$$

$$val(f) = f(s, a) + f(s, c) = 3 + 5 = 8$$

Rez sieťou

Nech $V = W \cup \overline{W}, W \cap \overline{W} = \emptyset$,
tak, že $s \in W \wedge t \notin \overline{W}$,
portom rezom $W, \overline{W}$,
nazveme množinu hrán
$\{(a,b) | a \in W \wedge b \in \overline{W}\}$



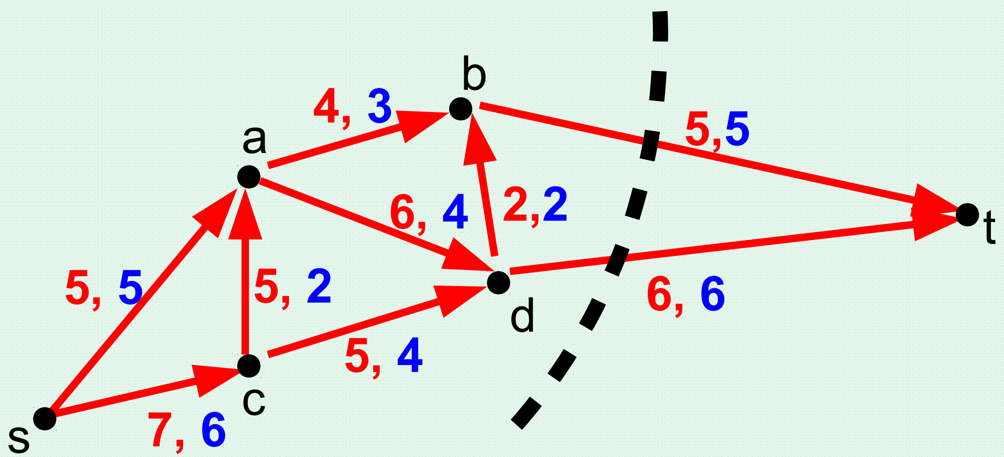
Kapacitou rezu potom nazývame

$$c(W, \overline{W}) = \sum_{\substack{v \in W \\ w \in \overline{W}}} c(v, w)$$

$$c(s,c) + c(a,b) + c(a,d) = 7 + 4 + 6 = 17$$

Ford-Fulkersonova veta

V každej sieti sa veľkosť maximálneho toku, rovná veľkosti minimálneho rezu.



$$val(f) = f(s,a) + f(s,c) = 5 + 6 = 11$$
$$c(V \setminus \{t\}, \{t\}) = c(b,t) + c(d,t) = 5 + 6 = 11$$

Tok zväčšujúca cesta

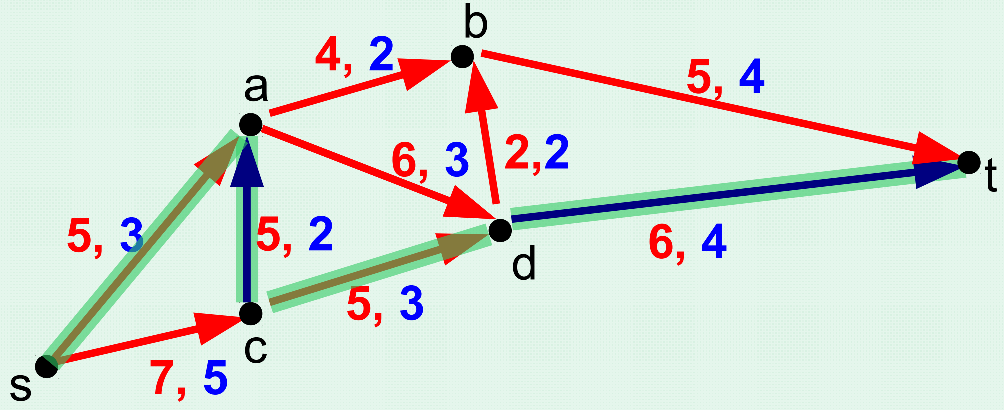
Zyšková kapacita hrany z tok zväčšujúcej cesty je

$\Delta(e) = c(e) - f(e)$, pre hrany dopredné (v_i, v_{i+1})

$\Delta(e) = f(e)$, pre hrany spätné (v_{i+1}, v_i)

Zvyšková kapacity cesty p je potom

$$\Delta(p) = min_{e \in p}\{\Delta e\}$$

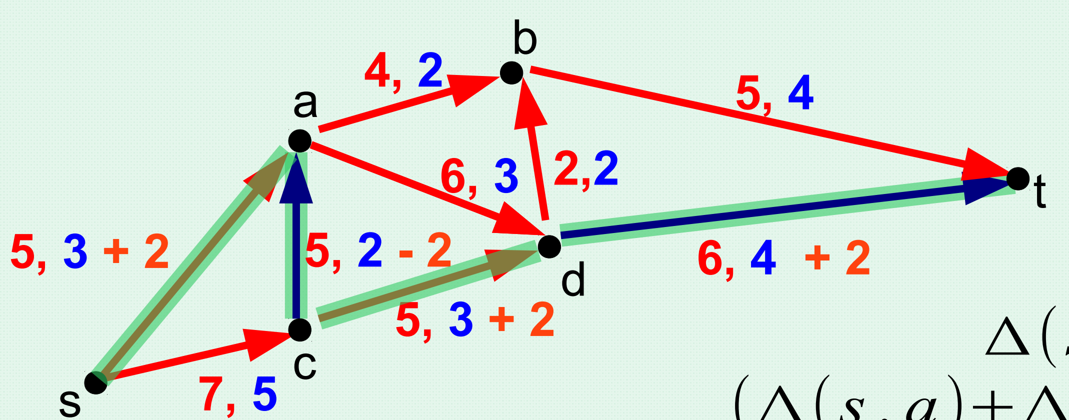


$$\Delta(s,a) = 5 - 3 = 2$$
$$\Delta(c,a) = 2$$
$$\Delta(c,d) = 5 - 3 = 2$$
$$\Delta(d,t) = 6 - 4 = 2$$
$$\Delta(p) = 2$$

Tok zväčšujúca cesta

Zýšením (znížením) toku na všetkých hranách z p , o $\Delta(p)$ dostanem nový tok g, pre kotrý platí

$$val(g) = val(f) + \Delta(p)$$



$$\Delta(s, a) + \Delta(s, c) = 3 + 5 = 8$$

$$(\Delta(s, a) + \Delta(p)) + \Delta(s, c) = (3 + 2) + 5 = 10$$

Ford Fulkersonov algoritmus

Kým existuje tok zväčšujúca cesta p, zýš tok o $\Delta(p)$

Veta: Ak sa veľkosť toku f rovná kapacite rezu $(W, \overline{W})$, potom f je maximálny tok a $(W, \overline{W})$ je minimálny rez v sieti N.

Veta: Nech f je tok a $(W, \overline{W})$ rez v sieti N.
Potom $val(f) = c(W, \overline{W})$ vtedy a len vtedy ak

a) $\forall \{e=(x,y) | x \in W \wedge y \in \overline{P}\}, \quad f(e) = c(e)$ a

b) $\forall \{e=(x,y) | x \in \overline{P} \wedge y \in P\}, \quad f(e) = 0$ a

Veta: f je maximálny tok v sieti N, vtedy a len vtedy, ak neexistuje tok zvyšujúca cesta

Ford Fulkersonov algoritmus

Zložitosť: $O(|E| max\, val(f))$

Problém: Pre reálne hodnoty kapacity a toku, algoritmus môže bežať do nekonečna a nemusí konvergovať k riešeniu

Zlepšenie:
Vyhľadávať tok rozširujúcu cestu ako najkratšiu, alebo s najvačšou zvyškovou kapacitou

Edmonds – Karp algoritmus

Zložitosť pri hľadaní najkratšej rozširujúcej cesty:

$$O(|V||E^2|)$$

Zložitosť pri hľadaní rozširujúcej cesty s najväčšou zvyškovou kapacitou: $O(|V^2||E|\ln(max.\ kapacita\ hrany))$

```
        break
    f := f + m
    (Backtrack search, and write flow)
    v := t
    while v ≠ s
        u := P[v]
        F[u,v] := F[u,v] + m
        F[v,u] := F[v,u] - m
        v := u
return (f, F)
```

```
    Q.push(s)
    while Q.size() > 0
        u := Q.pop()
        for v in E[u]
            (If there is available capacity, and v
is not seen before in search)
            if C[u,v] - F[u,v] > 0 and P[v] = -1
                P[v] := u
                M[v] := min(M[u], C[u,v] - F[u,v])
                if v ≠ t
                    Q.push(v)
                else
                    return M[t], P
    return 0, P
```

Edmonds – Karp algoritmus

```
algorithm EdmondsKarp
    input:
        C[1..n, 1..n] (Capacity matrix)
        E[1..n, 1..?] (Neighbour lists)
        s             (Source)
        t             (Sink)
    output:
        f             (Value of maximum flow)
        F             (A matrix giving a legal
flow with the maximum value)
    f := 0 (Initial flow is zero)
    F := array(1..n, 1..n) (Residual capacity
from u to v is C[u,v] - F[u,v])
    forever
        m, P := BreadthFirstSearch(C, E, s, t)
        if m = 0
            break
        f := f + m
        (Backtrack search, and write flow)
        v := t
        while v ≠ s
            u := P[v]
            F[u,v] := F[u,v] + m
            F[v,u] := F[v,u] - m
            v := u
    return (f, F)
```

```
algorithm BreadthFirstSearch
    input:
        C, E, s, t, F
    output:
        M[t]          (Capacity of path found)
        P             (Parent table)
    P := array(1..n)
    for u in 1..n
        P[u] := -1
    P[s] := -2 (make sure source is not
rediscovered)
    M := array(1..n) (Capacity of found path to
node)
    M[s] := ∞
    Q := queue()
    Q.push(s)
    while Q.size() > 0
        u := Q.pop()
        for v in E[u]
            (If there is available capacity, and v
is not seen before in search)
            if C[u,v] - F[u,v] > 0 and P[v] = -1
                P[v] := u
                M[v] := min(M[u], C[u,v] - F[u,v])
                if v ≠ t
                    Q.push(v)
                else
                    return M[t], P
    return 0, P
```

Dá sa to ešte lepšie?

Rok	Autor	Zložitosť
1957	Ford- Fulkerson	$O(\lvert E\rvert F)$
1972	Edmonds a Karp (najkratšia cesta)	$O(\lvert V\rvert \lvert E^2\rvert)$
1973	Edmons a Karp (najväčšia zyšková kapacita)	$O(\lvert V^2\rvert \lvert E\rvert \ln(C))$
1972	Dinic	$O(\lvert V^2\rvert \lvert e\rvert)$
1978	Malhotra, Kumar, Maheshwari algoritmus troch Indov	$O(\lvert V^3\rvert)$
1988	Goldberg	$O(\lvert V\rvert \lvert E\rvert \log(\lvert V\rvert / \lvert E^2\rvert))$

Baseball elimination

Otázka: Môže Harvard ešte vyhrať turnaj?

	w(i)	g(i)	g(i,j)			
Team	*Wins*	*To play*	*Y*	*H*	*C*	*B*
Yale	33	8		1	6	1
Harvard	29	4	1		0	3
Cornell	28	7	6	0		1
Brown	27	5	1	3	1	

Každá výhra znamená bod

Baseball elimination

- Harvard môže získať maximálne $W = 29+44 = 33$ bodov
- Harvard nebude eliminovaný zo súťaže ak
  - Brown nezíska viac ako $u(B) = W - w(B) = 33 - 27 = 6$ víťazstiev
  - Cornell nezíska viac ako $u(C) = W - w(C) = 33 - 28 = 5$ víťazstiev
  - Yale nezíska viac ako $u(Y) = W - w(Y) = 33 - 33 = 0$ víťazstiev

- Nech P je množina tímov iných ako Harvard: $P = \{Y, C, B\}$
- Nech Q je množina všetkých dvojíc z P

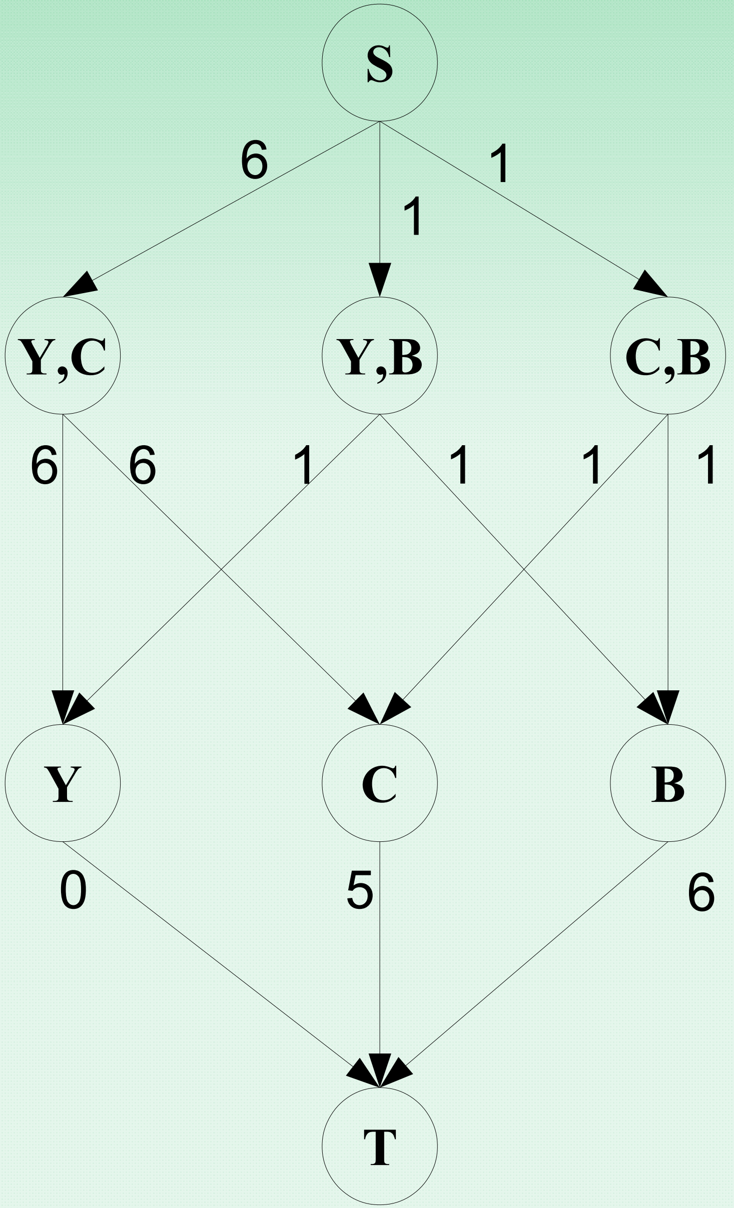
$$Q = \{(Y, C), (Y, B), (C, B)\}$$

- Počet hier, ktorý ešte ostáva odohrať medzi tímami z P

$$G = 6+1+1 = 8$$

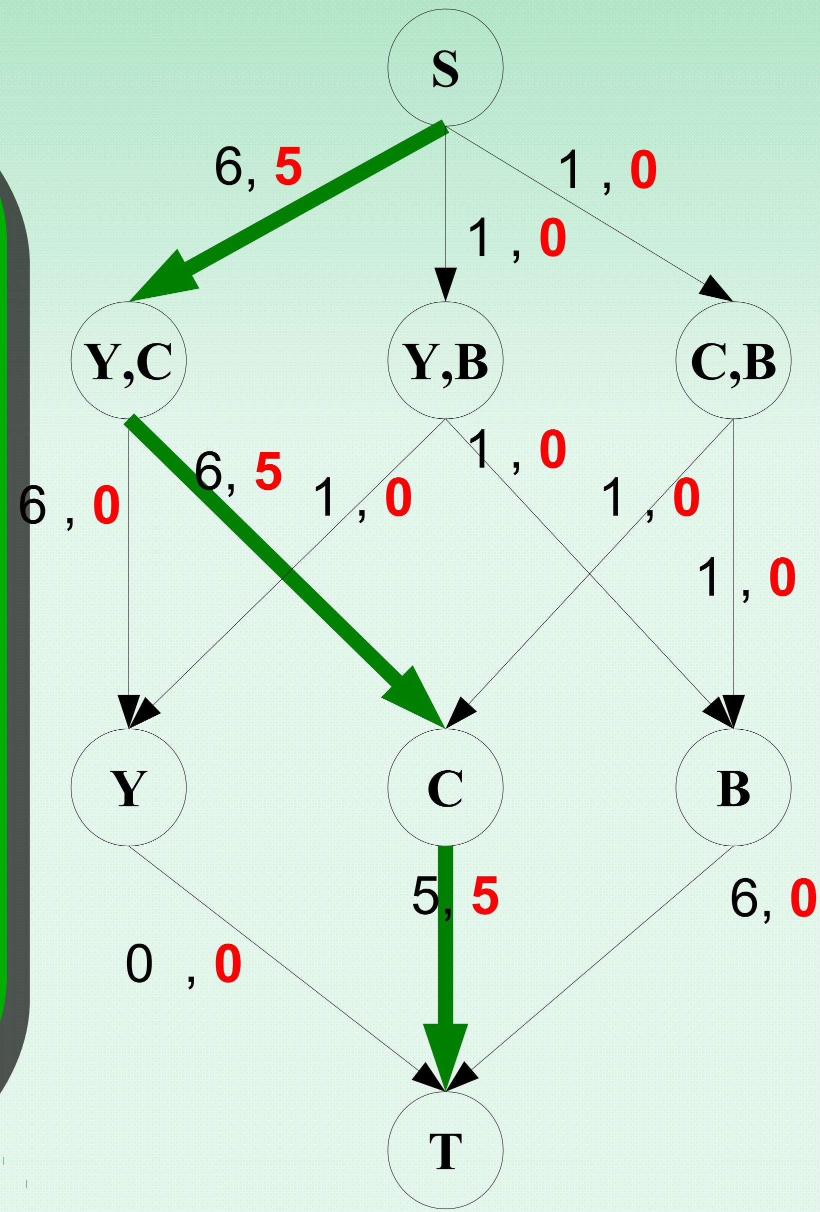
Baseball elimination

- Vytvoríme vrchol ***S***, v ktorom začnú všetky zápasy
- Vytvoríme vrchol, pre každý prvok ***(X,Y)*** z ***Q*** a hranu z ***S*** do ***(X,Y)*** s kapacitou rovnou počtu zápasov, ktoré majú ***X*** a ***Y*** odohrať
- Vytvoríme vrchol pre každý prvok z ***P*** hrany z každého vrchol ***X,Y*** do ***X*** a ***Y*** tak, že $c((X,Y),X) = c((X,Y),Y) = c(S,(X,Y))$
- Vytvoríme vrchol ***T*** a pridáme hrany z vrcholov z ***P*** do ***T*** s kapacitou ***u(X)***



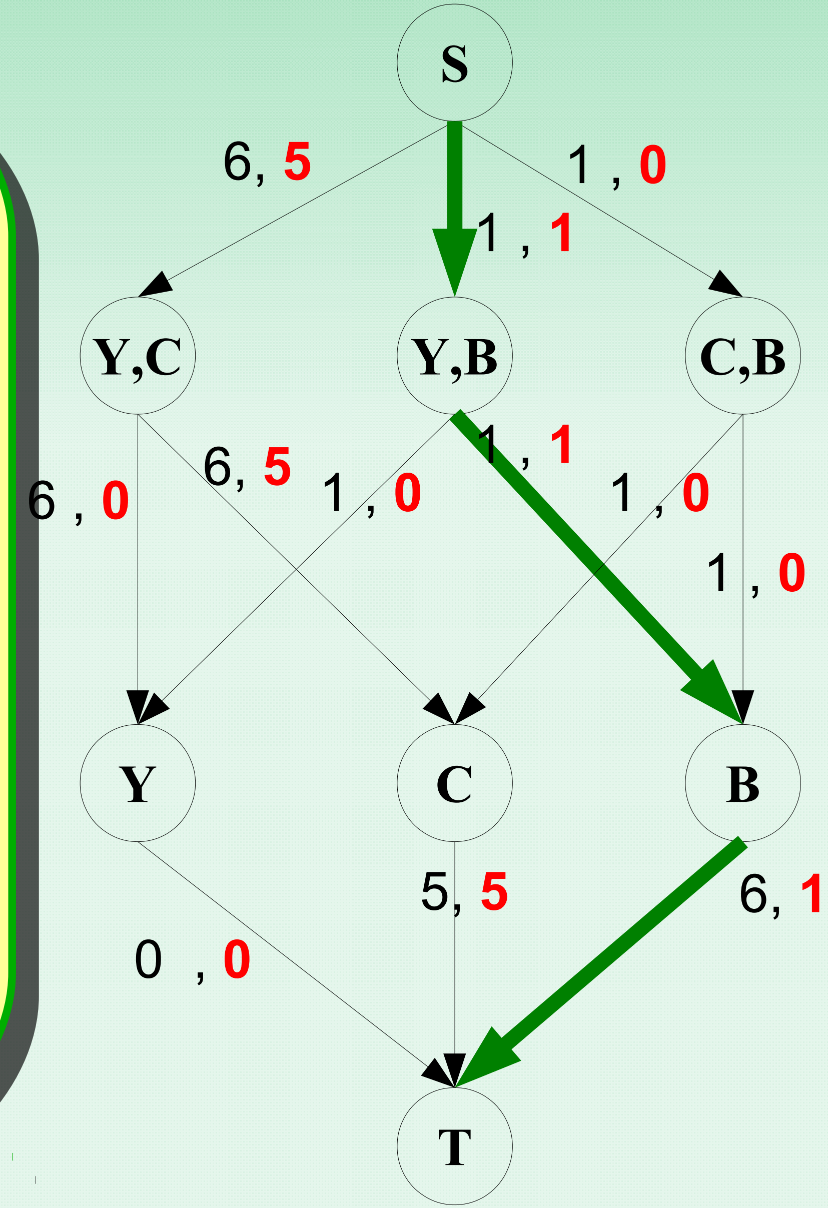
Baseball elimination

- Začneme s nulovým tokom
- Nájdeme tok zväčšujúcu cestu ***O*, (*Y*, *C*), *Y*, *T***, vďaka ktorej rozšírime tok o **5**



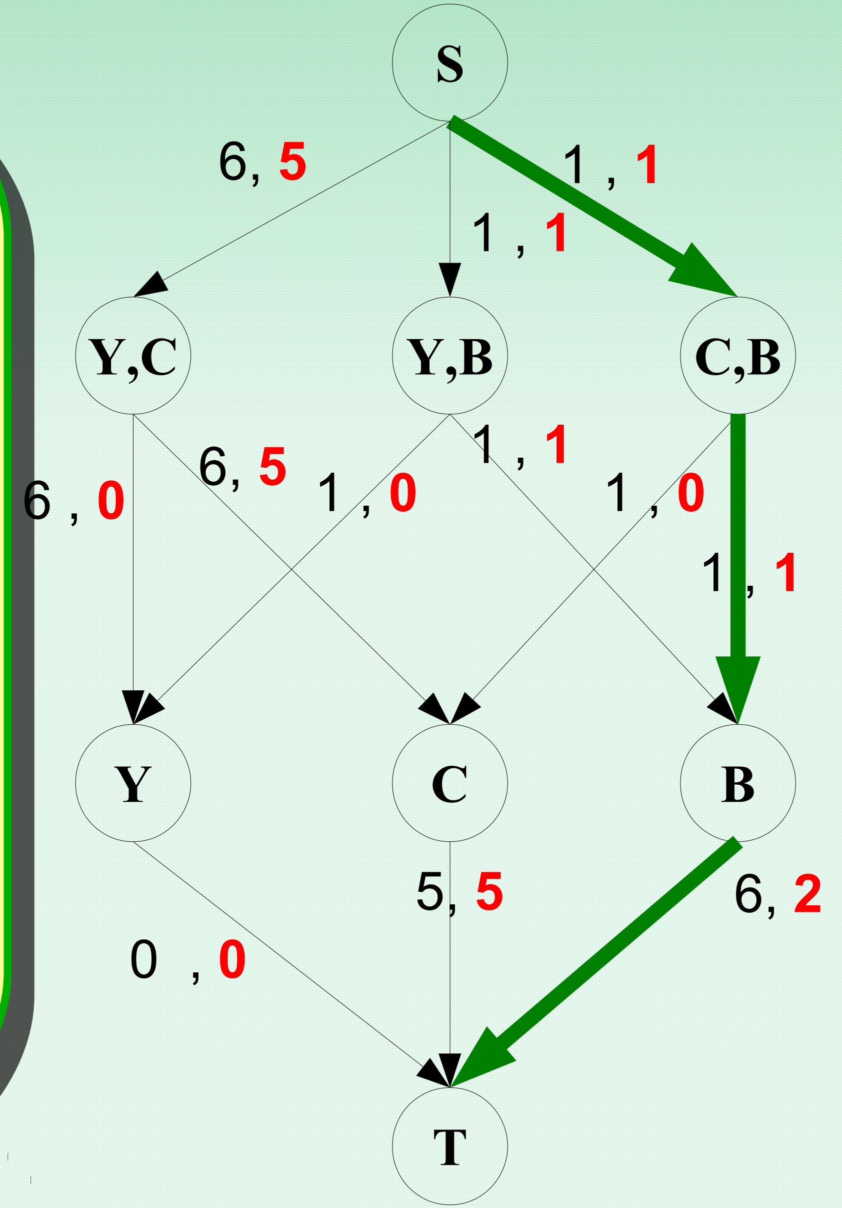
Baseball elimination

- Začneme s nulovým tokom
- Nájdeme tok zväčšujúcu cestu ***O*, (*Y*, *C*), C, *T***, vďaka ktorej rozšírime tok o **5**
- Nájdeme tok zväčšujúcu cestu ***O*, (*Y*, *B*), *Y*, *T***, vďaka ktorej rozšírime tok o **1**



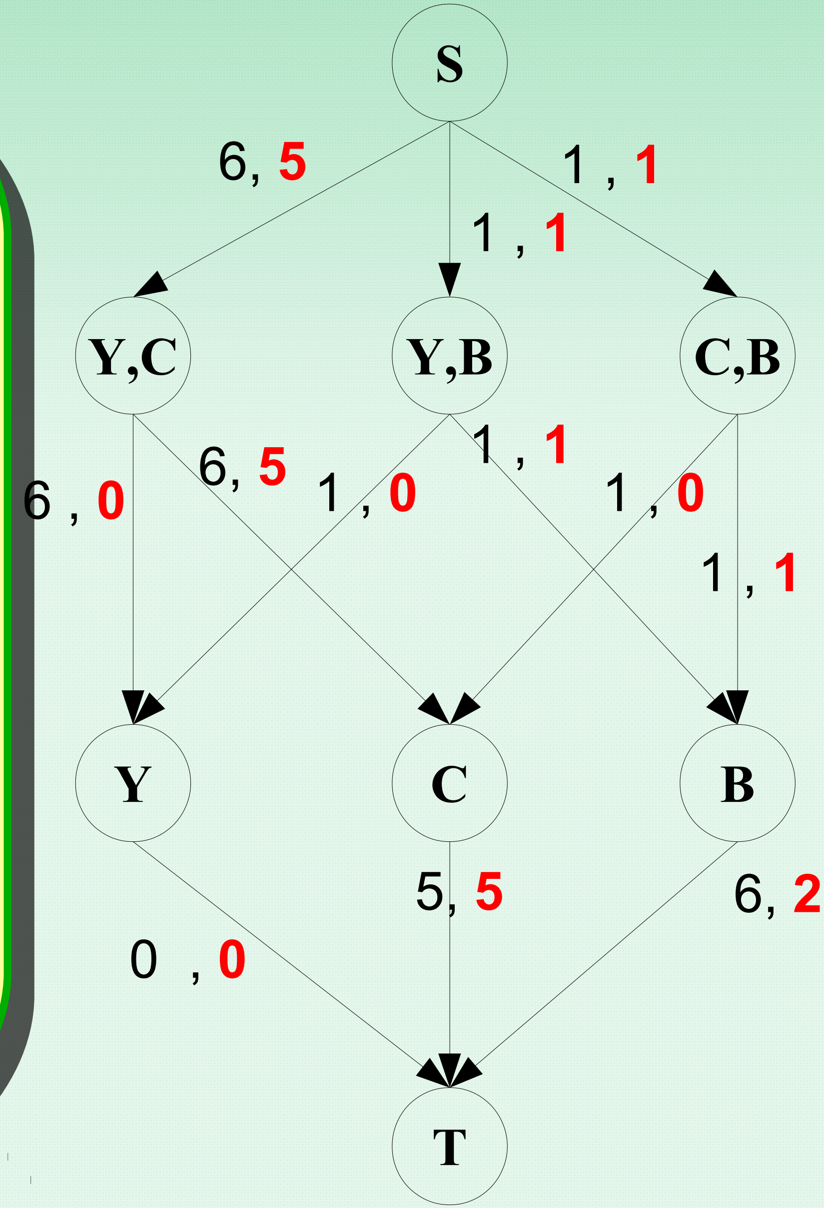
Baseball elimination

- Začneme s nulovým tokom
- Nájdeme tok zväčšujúcu cestu ***O*, (*Y*, *C*), *Y*, *T***, vďaka ktorej rozšírime tok o **5**
- Nájdeme tok zväčšujúcu cestu ***O*, (*Y*, *B*), B, *T***, vďaka ktorej rozšírime tok o **1**
- Nájdeme tok zväčšujúcu cestu ***O*, (C, *B*), B, *T***, vďaka ktorej rozšírime tok o **1**



Baseball elimination

- Začneme s nulovým tokom
- Nájdeme tok zväčšujúcu cestu ***O*, (*Y*, *C*), *Y*, *T***, vďaka ktorej rozšírime tok o **5**
- Nájdeme tok zväčšujúcu cestu ***O*, (*Y*, *B*), B, *T*,** vďaka ktorej rozšírime tok o **1**
- Nájdeme tok zväčšujúcu cestu ***O*, (C, *B*), B, *T*,** vďaka ktorej rozšírime tok o **1**
- Našli sme maximálny tok s hodnotou **7**



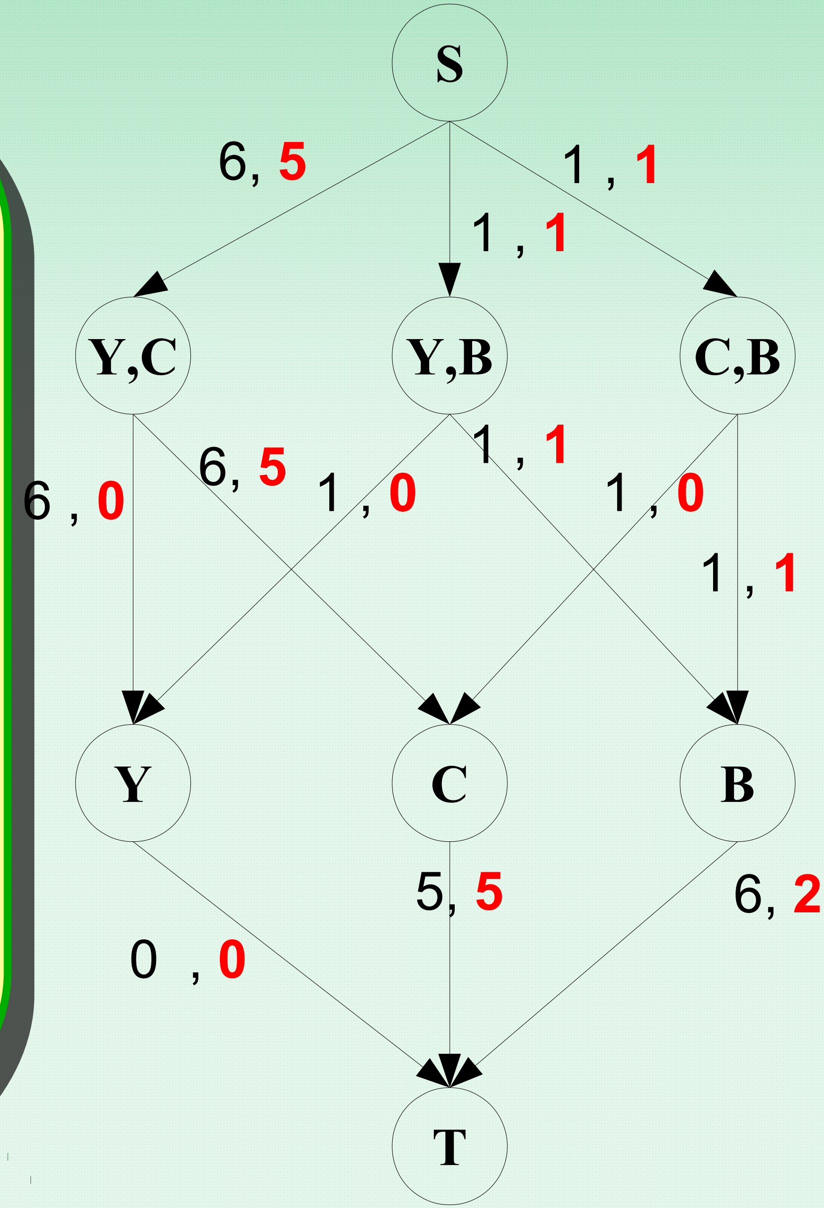
Baseball elimination

- Nájdi maximálny tok z *S* do *T*
- Ak je hodnota maximálneho toku = *G* , Harvard má stále ešte šancu na titul. V opačnom prípade už nemôže turnaj vyhrať

  max flow = 7 < 8 = G

  Harvard už turnaj nevyhrá

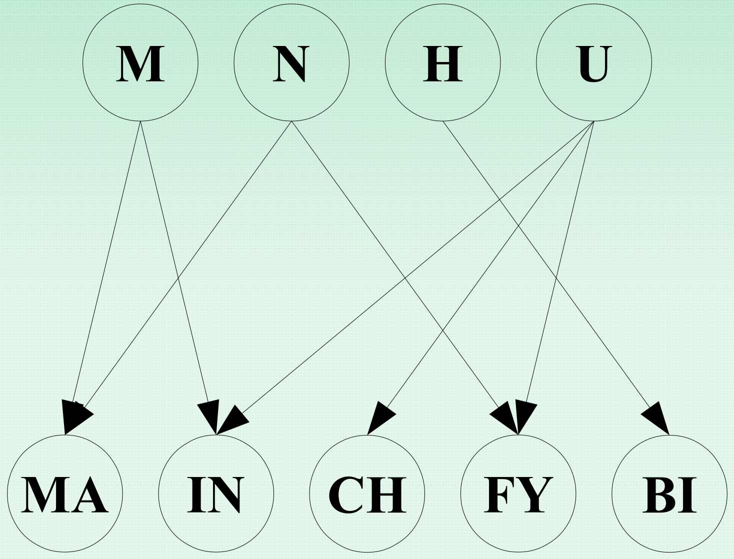
- **Vysvetlenie:** Ak sa maximálny tok rovná počtu zápasov ktorý ostáva odhrať, tak sa dá turnaj odhorať tak, aby žiadny z tímov *Y, C, B* nezískal viac ako *u(Y),u(C),u(B)* akbodov a tak Harvard by mal ešte šancu na výhru



Problém párovania

- Škola potrebuje zamestnať štyroch učiteľov, na vyučovanie matematiky, informatiky, chémie, fyziky a biológie.
- Prihlísili sa pán Malý na matematiku a informatiku, pán Novák na matematiku a fyziku, pani Hladná na biológiu a pani Úžasná na chémiu, fyziku a informatiku.
- **Môže škola prijať všetkých štyroch učiteľov, tak aby každý učil predmet o ktorý má záujem a žiadny dvana neučili ten istý predmet?**

(assignment problem)



Definícia problému

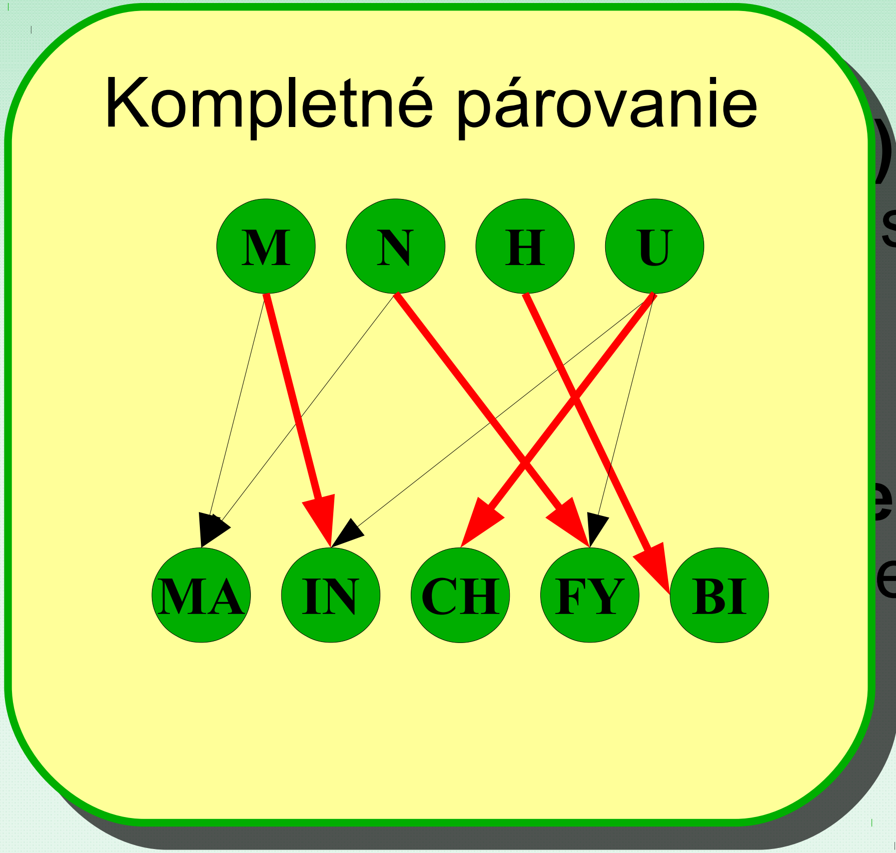
Nech $G = (V, E)$ je bipartitný graf, s partíciami $X \cup Y = V$.

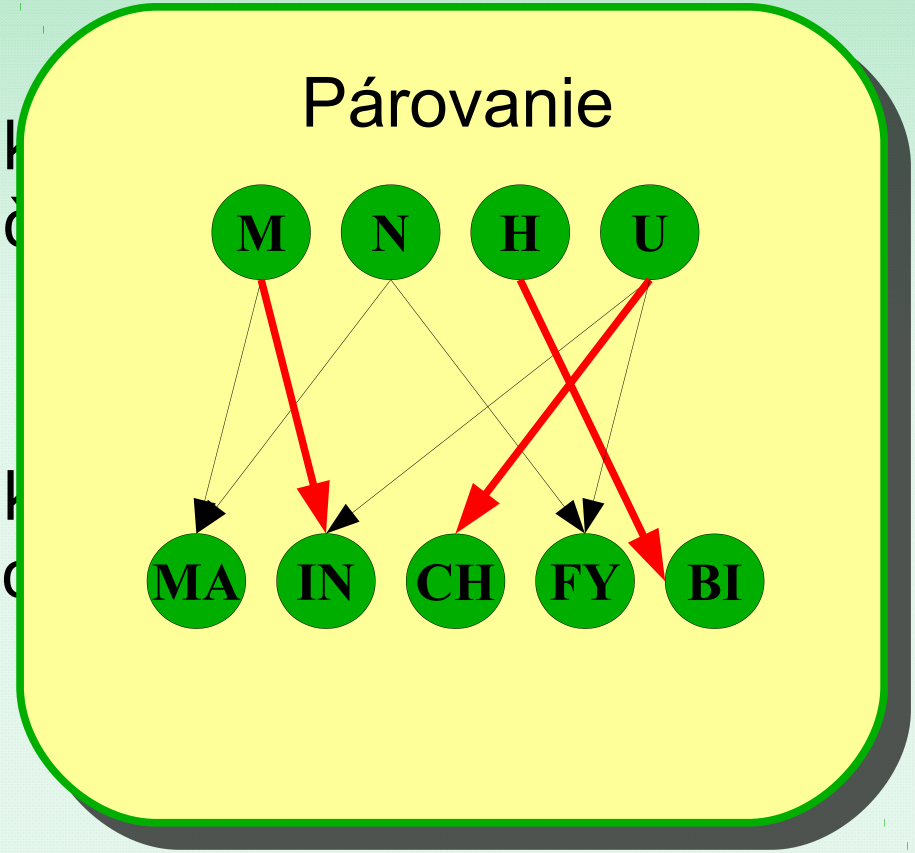
a) **Párovanie (matching)** je taká podmnožina hrán z E, že žiadne dve nemajú spoločný vrchol ani v množine X ani v množine Y.

b) **Kompletné párovanie** je také párovanie, že každý z vrcholov v množine X je koncovým vrcholom niektorej z hrán v párovaní.

Definícia problému

Nech $G = (V, E)$ je bipartitný graf, s partíciami $X \cup Y = V$.

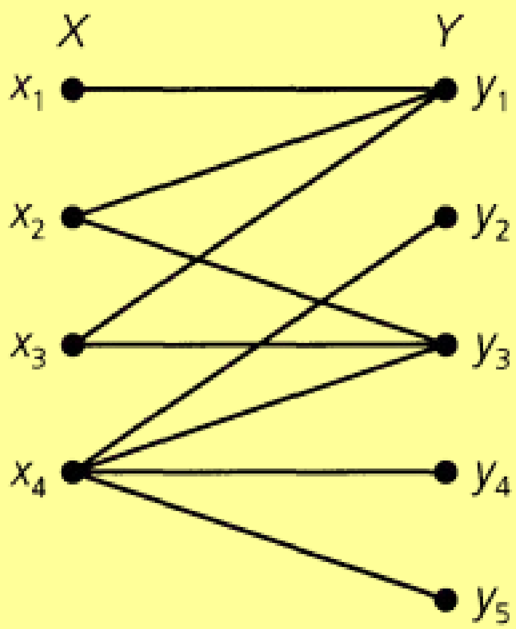




Problém párovania

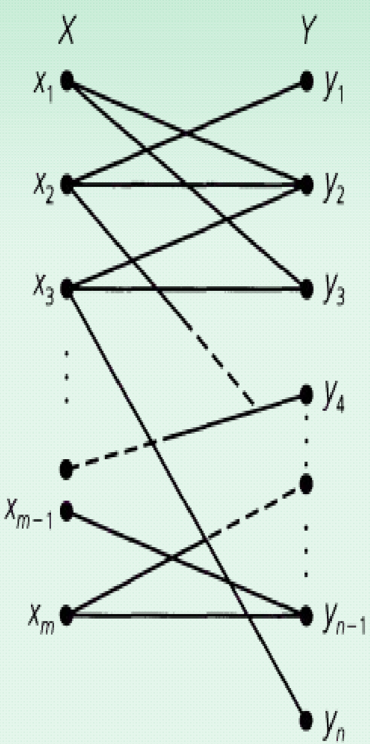
Philip Hall (1935): Marriage theorem

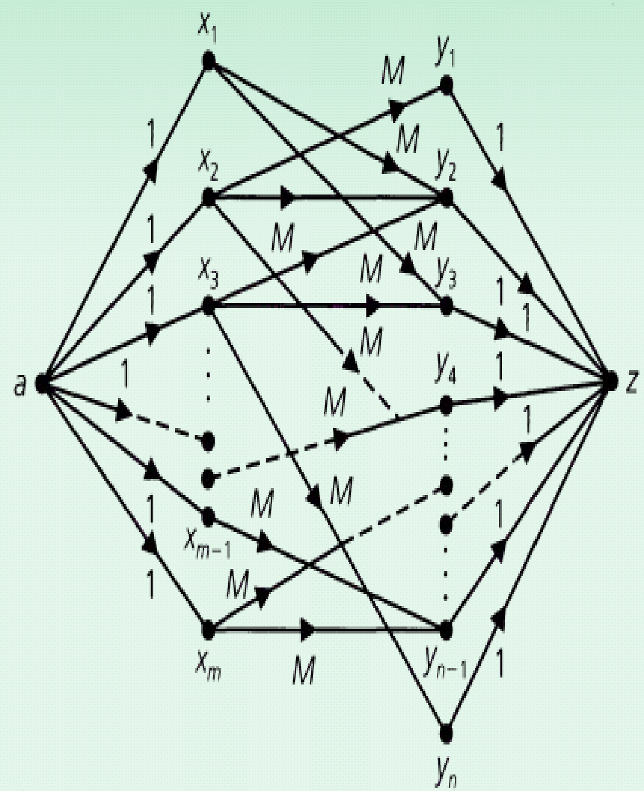
- Nech $G=(V, E)$ je bipartitný graf s partíciami $X \cup Y = V$. Kompletné párovanie X do Y existuje vtedy a len vtedy ak pre každú podmnožinu A množiny X platí $|A| \leq |R(A)|$, kde $R(A)$ je podmnožina Y, pozostávajúca s vrcholov incidentných s aspoň s jedným vrcholom v *A*.



$$A=\{x_1, x_2, x_3\}, \quad R(A)=\{y_1, y_3\}$$
$$|A|=3>2=|R(A)|$$

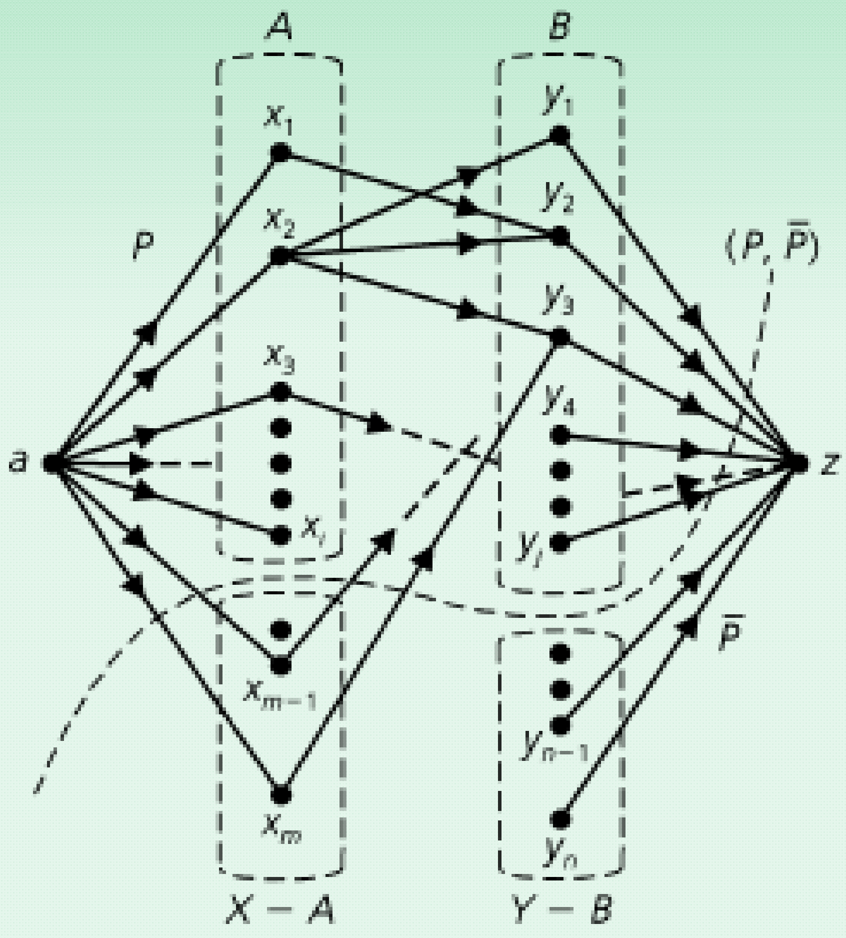
Problém párovania





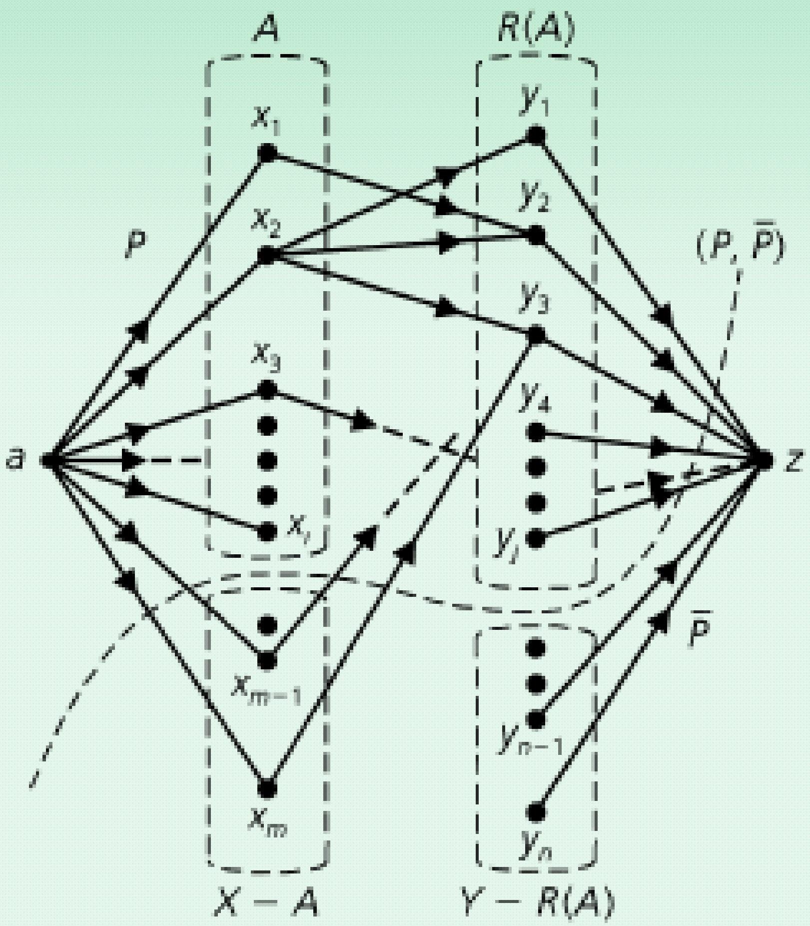
- Vytvor vrcholy a a z a spoj ich s množinou X (resp. Y)
- Novým hranám daj kapacitu 1, hranám z G, kapacitu $M \geq |X|$
- Kompletné párovanie existuje vtedy a len vtedy ak použije všetky hrany z vrchola a a veľkosť maximálneho toku je $|X|$

Problém párovania



- Ukážeme, že pre každý rez $c(P, \overline{P}) \geq |X|$
- Majme množiny $A = X \cap P$ a $B = Y \cap P$
- Ak existuje hrana z A do Y-B, tak jej kapacita je M a preto
- Inak rez pozostáva z hrán z a do X-A a z B do z, s kapacitou 1
- $c(P, \overline{P}) = |X - A| + |B| = |X| + |B| - |A|$ $B \supseteq R(A) \wedge |R(A)| \geq |A| \Rightarrow |B| \geq |A| \Rightarrow$ $c(P, \overline{P}) \geq |X|$

Problém párovania



- Inak ak $|A| > |R(A)|$, tak rez je daný hranami z a do X-A a z $R(A)$ do z
- $c(P, \overline{P}) = |X - A| + |R(A)| = |X| - (|A| - |R(A)|) < |X|$
- Potom podľa Ford-Fulkersonovej vety, takáto sieť má maximálny tok menší ako $|X|$ a teda pre daný problém neexistuje kompletné párovanie

Problém párovania

Dôsledok: Nech $G = (V, E)$ je bipartitný graf s partíciami X a Y, potom existuje kompletné párovanie X do Y, ak pre nejaké $k \in Z^+$ platí, $\forall x \in X, y \in Y \; deg(x) \geq k \geq deg(y)$

Definícia: Nech $G = (V, E)$ je bipartitný graf s partíciami X a Y, potom **maximálne párovanie** je párovanie ktoré zahŕňa čo najväčšiu podmnožinu množiny X.

Ak $A \subseteq X$, potom $\delta(A) = |A| - |R(A)|$ sa nazýva **nedostatkom** množiny A. Nedostatkom grafu G označujeme $\delta(G) = max\{\delta(A) | A \subseteq X\}$

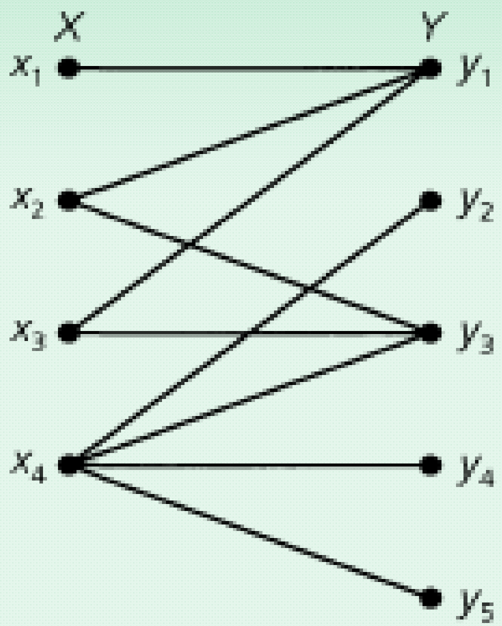
Problém párovania

Dôsledok: Nech $G = (V, E)$ je bipartitný graf s partíciami X a Y, potom existuje kompletné párovanie X do Y, ak pre nejaké $k \in Z^+$ platí, $\forall\, x \in X, y \in Y\ deg(x) \geq k \geq deg(y)$

Definícia: Nech $G = (V, E)$ je bipartitný graf s partíciami X a Y, potom **maximálne párovanie** je párovanie ktoré zahŕňa čo najväčšiu podmnožinu množiny X.

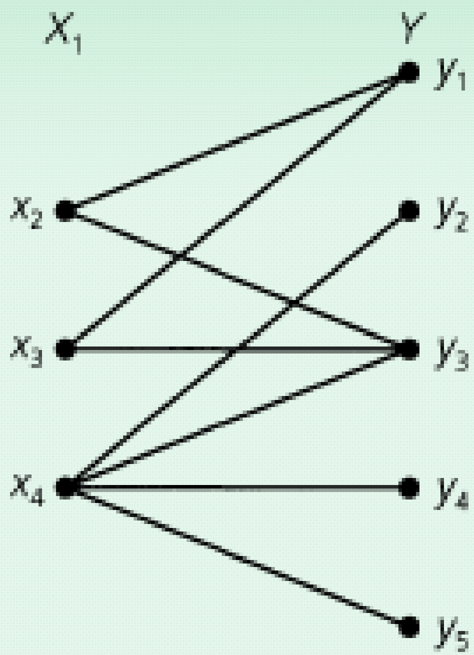
Ak $A \subseteq X$, potom $\delta(A) = |A| - |R(A)|$ sa nazýva **nedostatkom** množiny A. Nedostatkom grafu G označujeme $\delta(G) = max\{\delta(A) | A \subseteq X\}$

Problém párovania



- Graf na obrázku nemá kompletné párovanie.
- $A=\{x_{1,}x_{2,}x_3\}, \quad R(A)=y_{1,}Y_3$
  $\delta(A)=3-2=1, \quad \delta(G)=1$

Problém párovania



- Graf na obrázku nemá kompletné párovanie.
- $A=\{x_{1,}x_{2,}x_{3}\},\quad R(A)=y_{1,}Y_{3}$
  $\delta(A)=3-2=1,\quad \delta(G)=1$
- Odstránením jedného vrcholu, získame graf s kompletným párovaním, ktoré je zároveň maximálnym párovaním pre pôvodný graf

Problém párovania

Veta: Nech $G = (V, E)$ je bipartitný graf s partíciami X a Y, maximálny počet vrcholov z X, pre ktoré môžem nájsť párovanie je $|X| - \delta(G)$

Príklad: Nech $G = (V, E)$ je bipartitný graf s partíciami X a Y a $\forall\ x \in X, y \in Y\ deg(x) \geq 4 \wedge deg(y) \leq 5$. Nájdite horné ohraničenie pre $\delta(G)$ ak $|X| \geq 15$.

Nech $\emptyset \neq A \subseteq X$ a $E_1 \subseteq E$, kde $E_1 = \{(a, b) | a \in A, b \in R(A)\}$. Keďže $deg(a) \geq 4$, tak $|E_1| > 4|A|$ a pre $deg(b) < 5$, $|E_1| \leq 5|R(A)|$. Z $4|A| \leq 5|R(A)|$ dostávame $\delta(A) = |A| - |R(A)| \leq |A| - (4/5)|(A)|$
$= (1/5)|A|$. A keďže $A \subseteq X$ a $\delta(A) \leq (1/5)|X| = 3$. Následne $\delta(G) = max\{\delta(A) | A \subseteq X\} \leq 3$. A teda v danom grafe existuje maximálne párovanie veľkosti $|X| - 3$.

Ďalšie aplikácie

- data mining
- airline scheduling
- image processing
- network reliability
- distributed computing