

# Maksimalni protok kroz mrežu - Ford-Fulkerson, Edmonds-Karp

Marin Gunja

PMF-MO

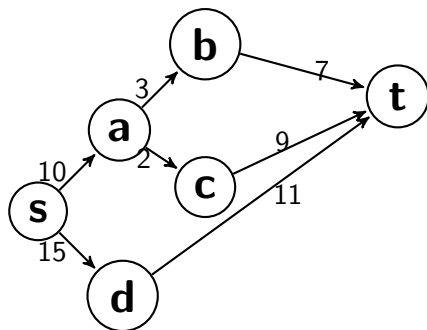
Seminar iz kolegija Oblikovanje i analiza algoritama

22.1.2019.

# Uvod - definicije

- Usmjereni graf:  $(V, E)$ ,  $V$  - vrhovi,  $E \subseteq V \times V$  - bridovi
- Kapacitete bridova dajemo s funkcijom  $c : V \times V \rightarrow \mathbb{R}^+$
- Imamo dva istaknuta vrha:  $s$  - izvor (*source*) i  $t$  - ponor (*sink*)
- *Protok* definiramo kao funkciju  $f : V \times V \rightarrow \mathbb{R}$  koja zadovoljava:
  - 1  $(\forall u, v \in V) f(u, v) \leq c(u, v)$  (ograničenje kapacitetom)
  - 2  $(\forall u, v \in V) f(u, v) = -f(v, u)$
  - 3  $(\forall u \in V \setminus \{s, t\}) \sum_{v \in V} f(u, v) = 0$
- Vrijednost protoka:  $|f| = \sum_{v \in V} f(s, v)$

# Primjeri



Moguće interpretacije:

- Cijevi (gledamo protok vode)
- Prijevoz (npr. kamioni)
- Internet

# Aplikacije i računala

*Zadatak.* Dano je  $n$  vrsta aplikacije; aplikacija  $k$ -te vrste ima  $a_k$ .

Dano je  $m$  računala, te za svaku vrstu aplikacije je dan popis računala koja mogu pokrenuti tu vrstu aplikacije. Računalo može pokrenuti samo jednu aplikaciju. Mogu li se sve aplikacije pokrenuti?

Modelirajte pomoću problema maksimalnog protoka.

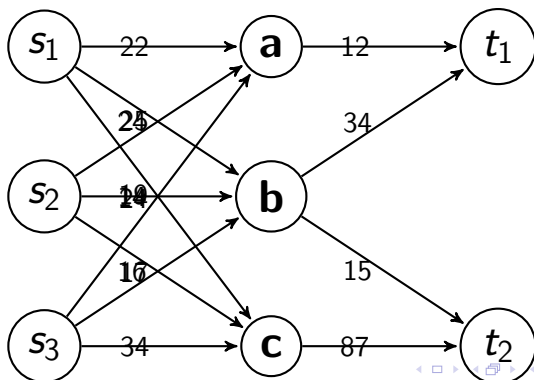
*Rješenje.* Vrste aplikacija i računala su čvorovi. Uvedemo još dva čvora:  $s$  (izvor) i  $t$  (ponor). Spajamo čvorove na sljedeći način:  $s$  spajamo sa  $k$ -tom vrstom aplikacije bridom kapaciteta  $a_k$ . Računala spajamo sa  $t$  bridovima kapaciteta 1. Vrste aplikacija spajamo sa računalima koja ih mogu pokrenuti bridovima kapaciteta  $\infty$ .

Kada je odgovor potvrđan? Onda i samo onda kada je maksimalni

tok jednak  $\sum_{k=1}^n a_k!$

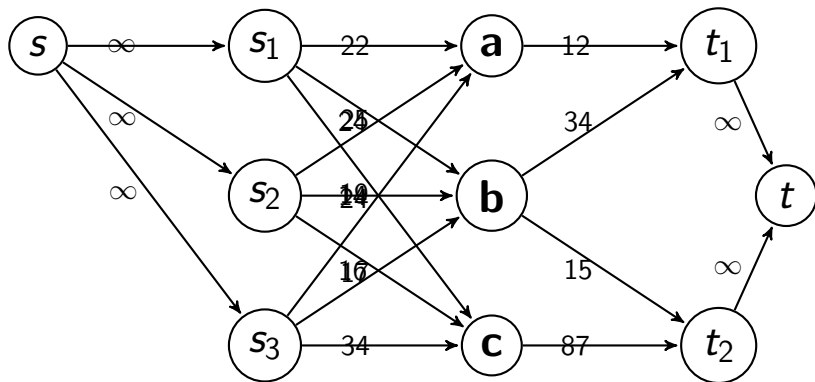
## Više izvora i ponora

- Kada imamo više izvora, uvedemo dodatan čvor kojeg spojimo sa svim izvorima bridovima kapaciteta  $\infty$  (nekim jako velikim brojem).
- Analogno ako imamo više ponora.



# Više izvora i ponora

- Prethodni graf postaje:



# Ford - Fulkersonova metoda

- Iterativna
- U početku protok postavljamo na 0 (tj.  $f \equiv 0$ ).
- *Povećavajući put* - put između  $s$  i  $t$  po kojem možemo poslati još toka
- Korak metode: sve dok postoji povećavajući put: povećaj protok (tj. povećaj  $f$ )
- Rezidualne težine:  $c_f(u, v) = c(u, v) - f(u, v)$
- Rezidualni graf:  $G_f = (V, E_f)$ , gdje je

$$E_f = \{(u, v) \in V \times V \mid c_f(u, v) > 0\}$$

# Ford - Fulkersonova metoda - matematički rezultati

**Lema 1.** Neka je  $G = (V, E)$ ,  $f$  protok u  $G$ . Neka je  $G_f$  rezidualni graf induciran s  $f$ . Neka je  $f'$  protok u  $G_f$ .

Tada je  $f + f'$  protok u  $G$ ,  $|f + f'| = |f| + |f'|$ .

- Povećavajući put je sada bilo koji put između  $s$  i  $t$  u  $G_f$
- *Rezidualni kapacitet puta  $p$* :  $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$



# Ford - Fulkersonova metoda - matematički rezultati

**Lema 2.** Neka je  $G = (V, E)$ ,  $f$  protok u  $G$ ,  $p$  put između  $s$  i  $t$  u  $G_f$ .

Definiramo  $f_p : V \times V \rightarrow \mathbb{R}$  s

$$f_p(u, v) = \begin{cases} c_f(p), & (u, v) \in p \\ -c_f(p) & (v, u) \in p \\ 0, & \text{inače} \end{cases}$$

Tada je  $f_p$  protok u  $G_f$ .

- Uporabom ovih lema dokazujemo važan teorem.

**Teorem 1.** Neka je  $G = (V, E)$ ,  $f$  protok u  $G$ ,  $p$  put između  $s$  i  $t$  u  $G_f$ . Neka je  $f_p$  definirana kao u Lemi 2. Definiramo  $f' : V \times V \rightarrow \mathbf{R}$  sa  $f' = f + f_p$ . Tada je  $f'$  protok u  $G$ ,  $|f'| = |f| + |f_p| > |f|$ .

# Osnovni Ford-Fulkersonov algoritam

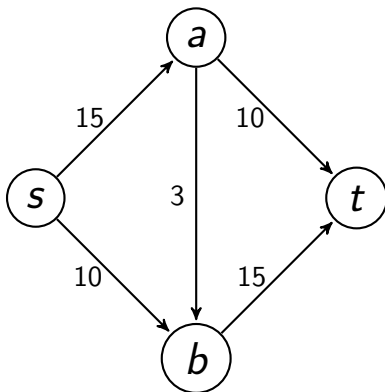
- Biramo BILO KOJI put između  $s$  i  $t$  u  $G_f$

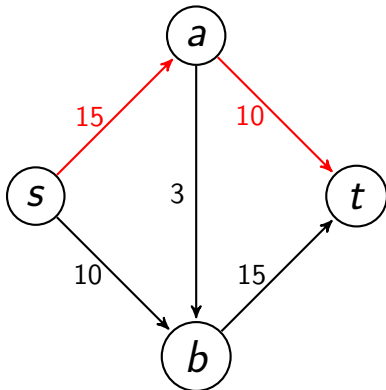
---

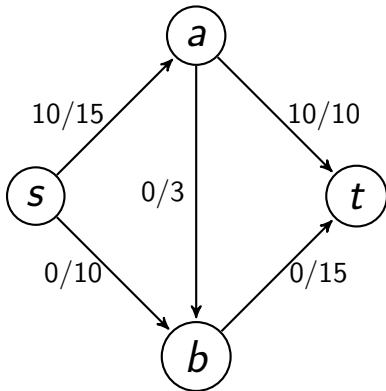
```
Ford-Fulkerson( $G$ ,  $s$ ,  $t$ )
for each edge  $(u, v)$  of  $G$ 
  do  $f[u, v] \leftarrow 0$ 
      $f[v, u] \leftarrow 0$ 
while there exists a path  $p$  from  $s$  to  $t$  in residual graph
  do  $c_f(p) \leftarrow \min \{c_f(u, v) \mid (u, v) \text{ on } p\}$ 
     for each edge  $(u, v)$  on  $p$ 
       do  $f[u, v] \leftarrow f[u, v] + c_f(p)$ 
           $f[v, u] \leftarrow -f[u, v]$ 
```

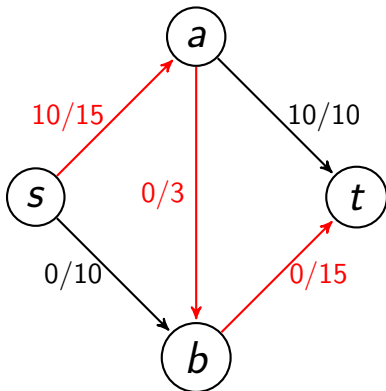
---

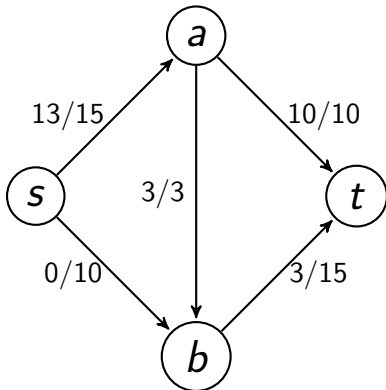
# Osnovni Ford-Fulkersonov algoritam - primjer

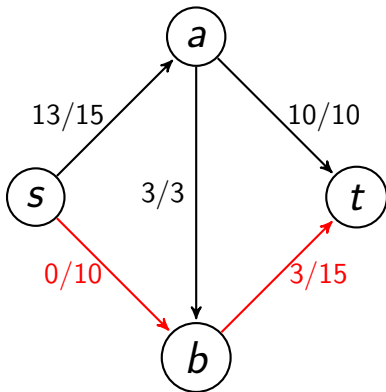




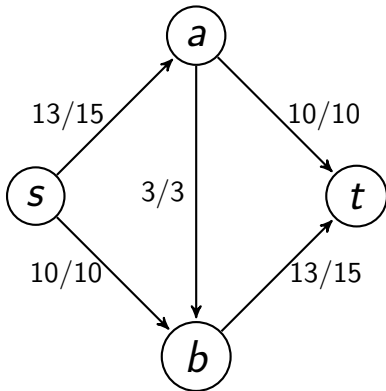








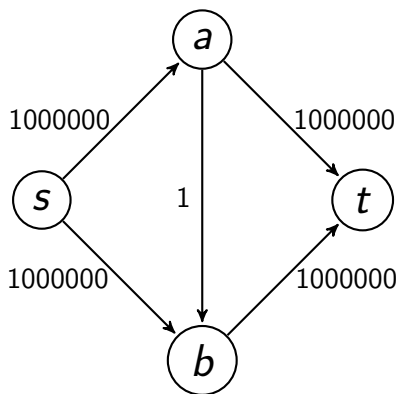


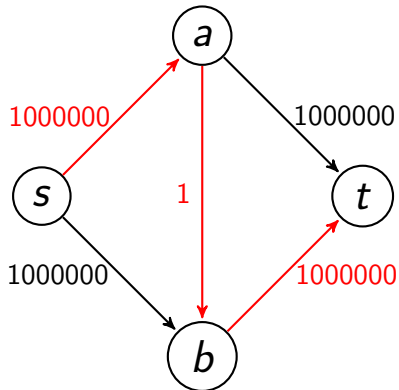


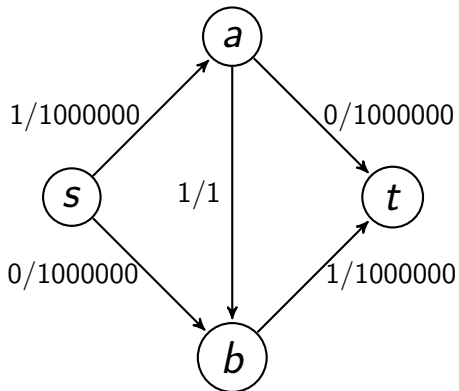
# Analiza osnovnog Ford-Fulkersonovog algoritma

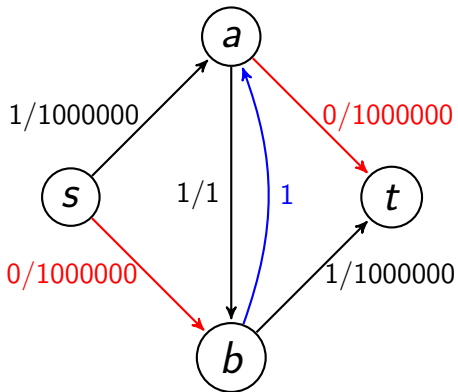
- Vrijeme izvršavanja jako ovisi o izboru puta povećavajućeg puta (u testovima koristimo DFS)
- Ako su kapaciteti cijeli brojevi, onda je vremenska složenost  $O(|E|f^*)$ , gdje je  $|E|$  broj bridova, a  $f^*$  maksimalni protok
- Inicijalizacija  $f$ :  $\Theta(|E|)$
- Protok se uvećava barem za 1; while se izvršava najviše  $f^*$  puta
- Za naći povećavajući put nam treba najviše  $O(|E|)$  (DFS ili BFS)

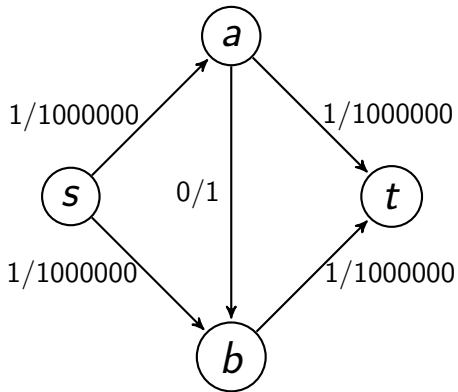
# Osnovni Ford-Fulkersonov algoritam - loša situacija











... i tako dalje...

# Edmonds-Karp algoritam

- Poboljšanje osnovnog Ford-Fulkersonovog algoritma
- Vremenska složenost:  $O(|V||E|^2)$
- Najkraći put između  $u$  i  $v$  u  $G_f$  definiramo sa  $\delta_f(u, v)$
- Edmonds-Karp bira najkraći put između  $s$  i  $t$  u  $G_f$ :  $\delta_f(s, t)$
- Prirodno je koristiti BFS.



# Edmonds-Karp - analiza složenosti

**Lema 3.** Neka je  $G = (V, E)$ , neka su  $s$  izvor i  $t$  ponor. Koristeći Edmonds-Karp algoritam,  $(\forall u, v \in V \setminus \{s, t\}) \delta_f(u, v)$  se monotono povećava sa svakom iteracijom (svakim povećanjem toka).

**Teorem 2.** Edmonds-Karp algoritam koristi  $O(|V||E|)$  povećanja protoka.

**Teorem 3.** Vremenska složenost Edmonds-Karm algoritma je  $O(|V||E|^2)$ .

**Dokaz.** Svaka iteracija Edmonds-Karp algoritma se može implementirati u  $O(|E|)$  vremenskoj složenosti. Po Teoremu 2. imamo  $O(|V||E|)$  iteracija.

# Usporedba DFS Ford-Fulkersonov algoritma i Edmonds-Karpovog algoritma ( $|V| = 10$ ) u $\mu s$

FF	EKA
16	22
9	30
9	29
9	28
8	22
9	35
10	28
12	29
11	35
9	24

FF	EKA
8	22
9	29
8	25
7	20
9	25
9	36
10	28
8	30
9	31
12	33

FF	EKA
10	34
9	31
10	27
10	37
11	40
10	36
9	33
10	31
11	32
9	25

# Usporedba DFS Ford-Fulkersonov algoritma i Edmonds-Karpovog algoritma ( $|V| = 20$ ) u $\mu s$

FF	EKA
58	120
70	150
60	100
50	120
40	130
60	160
60	120
50	120
50	110
60	130

FF	EKA
60	100
70	140
60	150
60	135
53	134
50	130
50	130
50	100
50	90
50	130

FF	EKA
70	111
40	90
40	140
50	110
40	90
50	143
40	110
50	80
60	80
60	90

# Usporedba DFS Ford-Fulkersonov algoritma i Edmonds-Karpovog algoritma ( $|V| = 50$ ) u $\mu s$

FF	EKA	FF	EKA	FF	EKA
695	900	850	900	700	750
800	1000	850	1000	850	950
750	900	900	850	800	900
750	900	850	900	750	950
750	1000	750	750	750	900
750	1050	750	850	850	900
800	1000	750	900	750	850
800	1000	850	1050	800	850
750	950	800	750	750	950
800	950	800	950	800	950

# Usporedba DFS Ford-Fulkersonov algoritma i Edmonds-Karpovog algoritma ( $|V| = 100$ ) u ms

FF	EKA	FF	EKA	FF	EKA
6.89	5.00	7.40	6.52	6.90	6.00
7.00	6.00	11.02	7.32	7.40	6.00
7.10	5.50	9.40	5.20	7.00	5.90
6.80	7.60	6.80	7.10	7.80	5.40
7.60	6.00	8.80	6.82	8.50	5.60
8.70	6.10	11.06	7.72	7.00	6.10
7.90	7.10	7.50	6.50	7.50	6.67
7.20	7.22	8.10	6.20	9.08	5.71
9.00	6.00	7.50	5.90	7.39	5.87
8.40	7.36	7.40	6.10	7.58	7.98

# Usporedba DFS Ford-Fulkersonov algoritma i Edmonds-Karpovog algoritma ( $|V| = 200$ ) u ms

FF	EKA
65.10	37.80
75.20	42.10
71.00	44.74
74.00	41.70
69.72	42.50
73.50	42.50
69.30	39.00
66.10	46.61
77.69	46.31
92.70	45.14

FF	EKA
72.20	40.40
64.70	38.40
72.10	42.90
80.44	44.22
88.48	59.68
82.10	48.38
80.72	46.12
81.70	43.20
83.48	43.00
66.90	39.80

FF	EKA
71.04	38.62
79.28	42.82
71.80	42.32
86.06	46.00
74.10	43.01
106.14	60.16
76.00	42.90
75.80	38.70
74.02	41.93
102.58	41.62

# Usporedba DFS Ford-Fulkersonov algoritma i Edmonds-Karpovog algoritma ( $|V| = 500$ ) u s

FF	EKA
1.42	0.55
1.53	0.75
2.17	0.79
1.64	0.59
1.79	0.68
1.52	0.60
1.61	0.62
1.47	0.55
1.46	0.55
1.43	0.54

FF	EKA
1.90	0.67
1.89	0.66
1.74	0.70
1.56	0.72
1.72	0.56
1.73	0.70
1.65	0.62
1.63	0.60
1.58	0.61
1.79	0.66

FF	EKA
2.42	0.65
1.91	0.79
1.58	0.54
1.35	0.51
1.48	0.70
1.63	0.64
1.71	0.66
1.62	0.70
1.84	0.59
1.41	0.54

# Usporedba DFS Ford-Fulkersonov algoritma i Edmonds-Karpovog algoritma ( $|V| = 700$ ) u s

FF	EKA
5.31	1.55
4.93	1.55
5.45	1.55
4.91	1.58
4.56	1.43
4.23	1.49
4.30	1.47
4.44	1.48
4.56	1.52
4.62	1.51

FF	EKA
4.41	1.43
4.33	1.47
4.50	1.43
4.39	1.46
4.34	1.44
4.27	1.44
4.34	1.50
4.44	1.47
4.46	1.38
4.96	1.80

FF	EKA
4.79	1.64
4.72	1.58
4.87	1.80
4.77	1.89
4.70	1.56
4.43	1.49
4.62	1.40
4.56	1.48
4.66	1.44
4.43	1.51



# Komentari

- Max flow - min cut
- Goldbergov algoritam:  $O(|V|^2|E|)$
- *The lift-to-front algorithm*:  $O(|V|^3)$

# Literatura

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 2nd edition, MIT Press, Massachusetts, 2001., 579-600
- M. H. Alsuwaiyel, Algorithms: Design Techniques and Analysis, 1999., Publishing House of Electronics Industry, 419-425
- S. Halim, F. Halim, Competitive Programming 3 - The New Lower Bound of Programming Contests, 2013., 163-170

# Pitanja?