Toky v sieťach

Sieť

Definition

Sieť S je štvorica $S = (G, z, s, c)$, kde

- G je orientovaný graf bez slučiek,
- $z \in V(G)$ je zdroj,
- $s \in V(G)$ je spotrebič,
- $c : E(G) \to R^+$ je kapacita.

Tok v sieti

Definition

Tok f v sieti $S = (G, z, s, c)$ je taká funkcia $f : E(G) \to R$, že

- $0 \leq f(e) \leq c(e), \quad \forall e \in E(G)$,
- $\sum_{Init(e)=v} f(e) = \sum_{Term(e)=v} f(e), \quad \forall v \in V(G), v \neq z, v \neq s$.

Úloha

Pre danú sieť S nájsť maximálny tok.

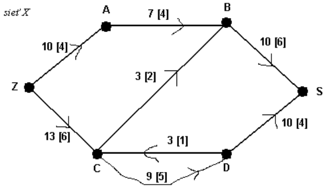
Zlepšujúca cesta

Definition

Cesta $z = v_0, v_1, v_2, \ldots, v_n = s$ sa nazýva zlepšujúca pre tok f, ak:

- môžeme zvýšiť tok vo všetkých hranách tejto cesty, ktoré sú súhlasne orientované s cestou,
- môžeme znížiť tok vo všetkých hranách tejto cesty, ktoré sú nesúhlasne orientované s cestou,

Zlepšujúca cesta - príklad



Zlepšujúce cesty:

- Z, A, B, S
- Z, C, B, S
- Z, C, D, S (2 cesty)
- Z, A, B, C, D, S (2 cesty)

Zlepšujúca cesta - princíp

Princíp:

- hodnota, o ktorú zvýšime tok, musí byť rovnaká pre všetky hrany
- súhlasnosť orientácie závisí od cesty

Ako nájsť zlepšujúcu cestu?

Nájdenie zlepšujúcej cesty

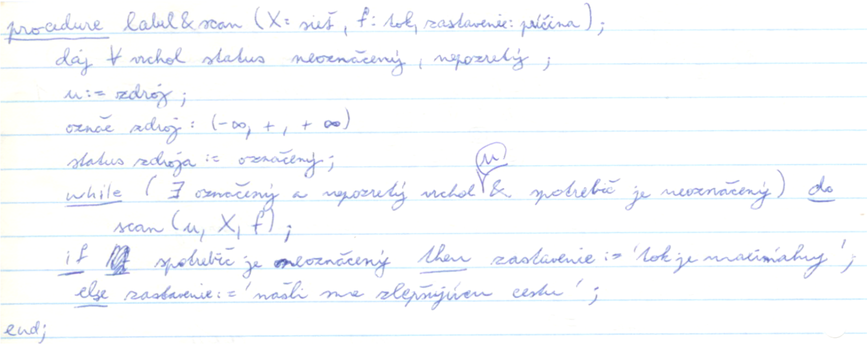
Definition

Majme hranu $< u, v >$. Potom hrana $< u, v >$ je použiteľná pre cestu z u do v, ak $f(< u, v >) < c(< u, v >)$. Podobne hrana $< v, u >$ je použiteľná pre cestu z u do v, ak $f(< v, u >) > 0$.

Nájdenie zlepšujúcej cesty - scan

```
procedure scan (u: vrchol, X: sieť, f: tok);
  for ∀ neoznačený vrchol spojený s vrcholom u hranou <u,v> alebo <v,u>
    do if f(<u,v>) < c(<u,v>) then
           označíme v: (u, +, min(z(u), c(<u,v>) - f(<u,v>)));
                                 to, čo do u priteká
       else if f(<v,u>) > 0 then
           označíme v: (u, -, min(z(u), f(<v,u>)));
end;
```

Obrázok: Procedúra scan.



```
procedure label&scan (X: sieť, f: tok, zastavenie: príčina);
    daj ∀ vrchol status neoznačený, nepozretý;
    u := zdroj;
    označ zdroj: (−∞, +, +∞)
    status zdroja := označený;
    while (∃ označený a nepozretý vrchol (u) & spotrebič je neoznačený) do
        scan(u, X, f);
    if spotrebič je neoznačený then zastavenie := 'tok je maximálny';
    else zastavenie := 'našli sme zlepšujúcu cestu';
end;
```

Obrázok: Procedúra label and scan.

```
procedure zv_toku (X: sieť; f: tok; množstvo: real);
  { bola vykonaná procedúra label & scan };
  v := spotrebič;
  množstvo := z(spotrebiča);
  repeat
     (previous, sign, z) := label(v);
     if (sign = '+') then f(<previous, v>) := f(<previous, v>) + množstvo;
     else f(<v, previous>) := f(<v, previous>) - množstvo;
     v := previous;
  until v = source;
end;
```

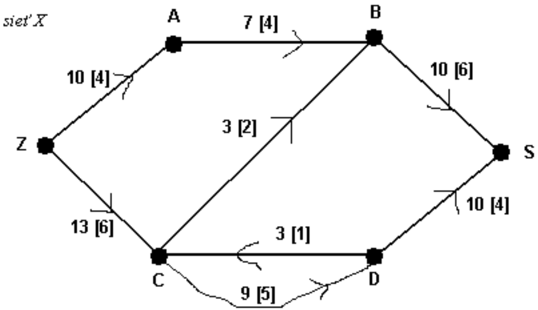
Obrázok: Procedúra zväčšenie toku.

Fordov-Fulkersonov algoritmus

```
procedure Ford-Fulkerson (X: sieť; f: tok; maxtok: real);
  { nájde maxtok pre sieť X };
  f := ∅;          { pre každú hranu siete X };
  maxtok := ∅;
  repeat
      label & scan (X, f, zastavenie)
      if zastavenie = 'našli sme zlepšujúcu cestu' then zv-toku (X, f, množstvo);
      maxtok := maxtok + množstvo;
  until zastavenie = 'tok je maximálny';
end;
```

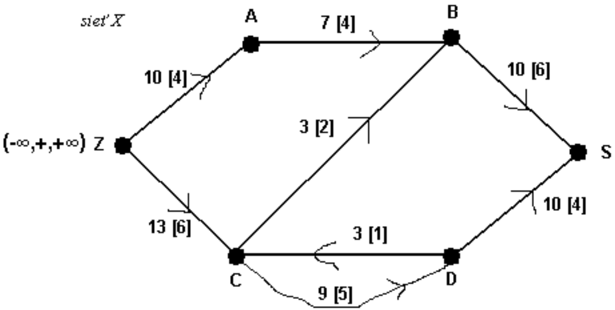
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



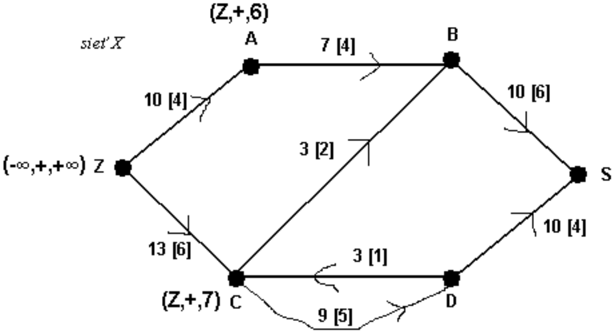
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



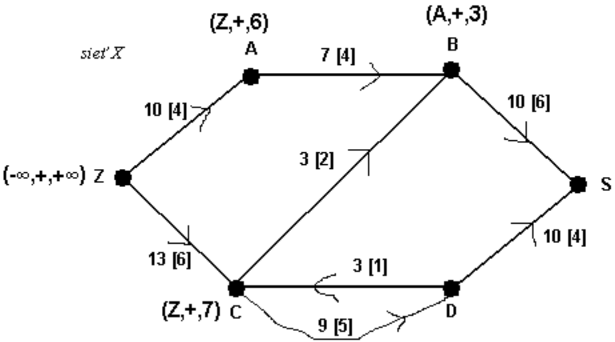
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad

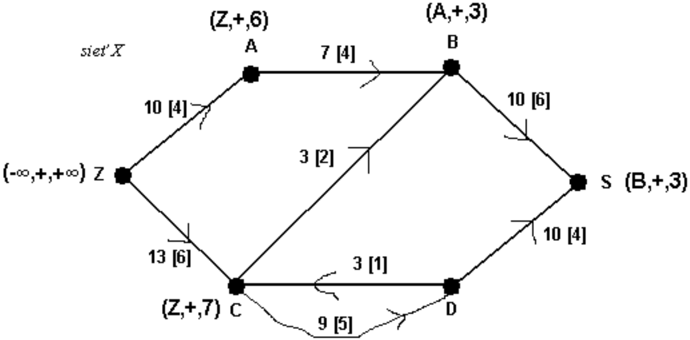


Obrázok: Fordov-Fulkersonov algoritmus.

Príklad

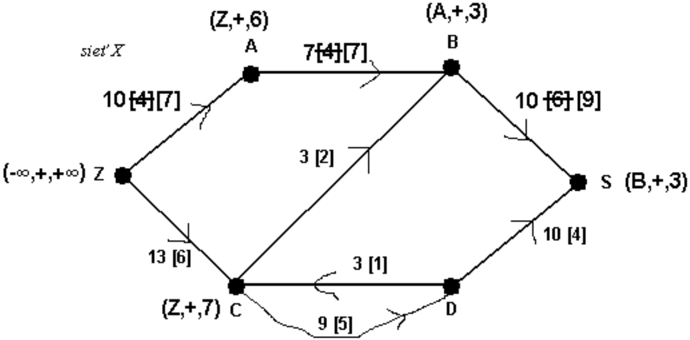


Obrázok: Fordov-Fulkersonov algoritmus.



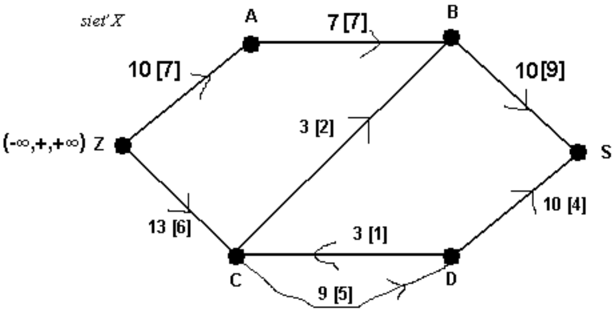
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



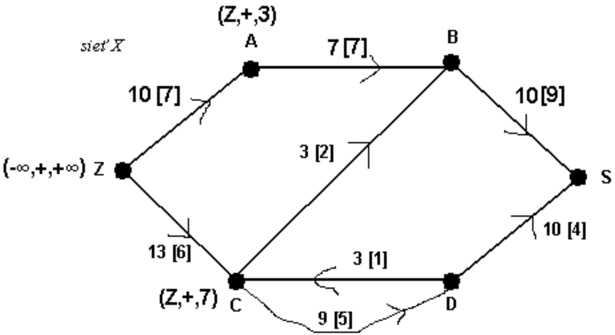
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



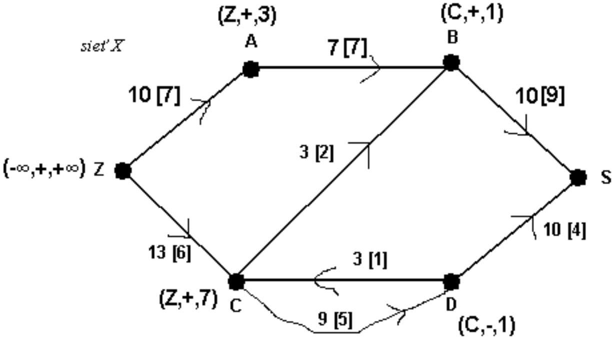
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



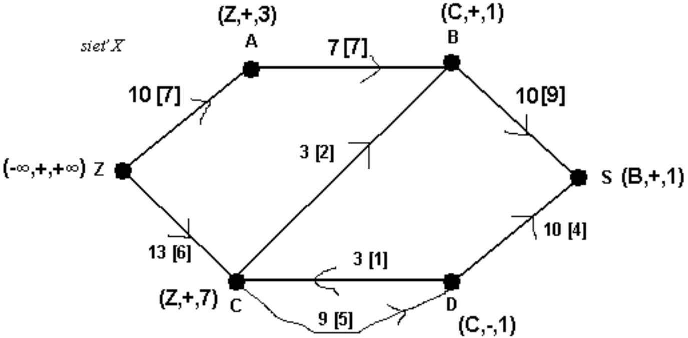
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



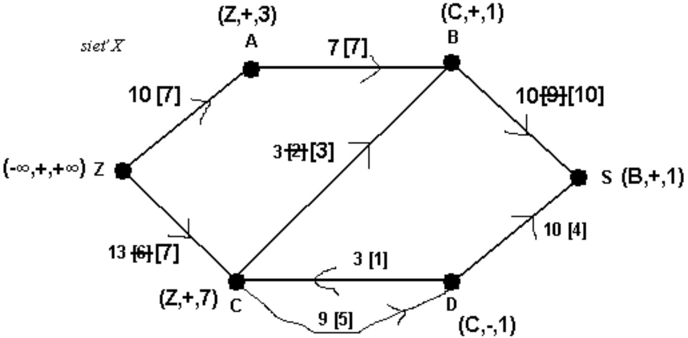
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



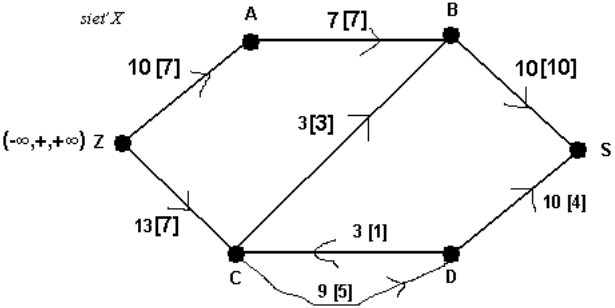
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



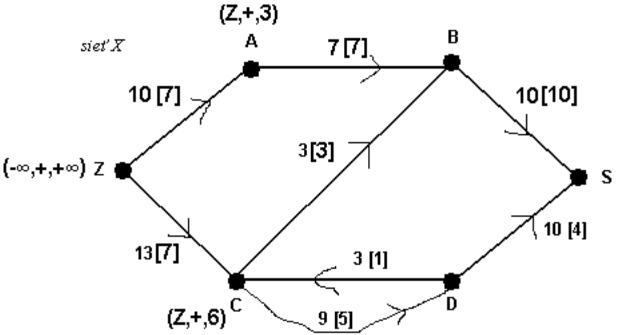
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



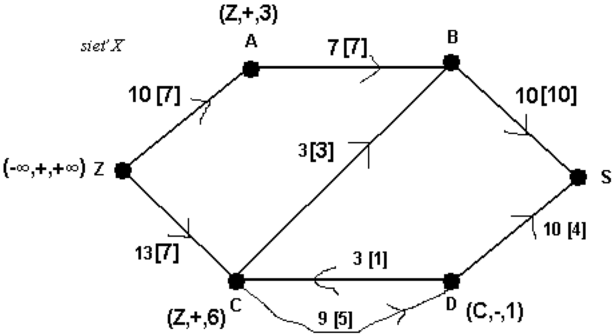
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



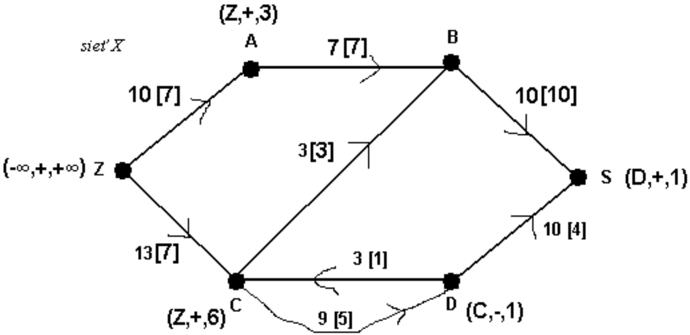
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



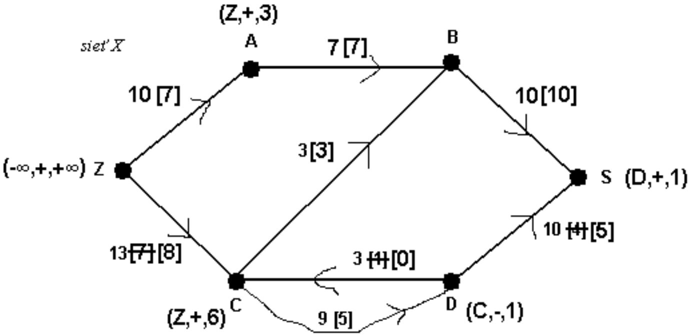
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



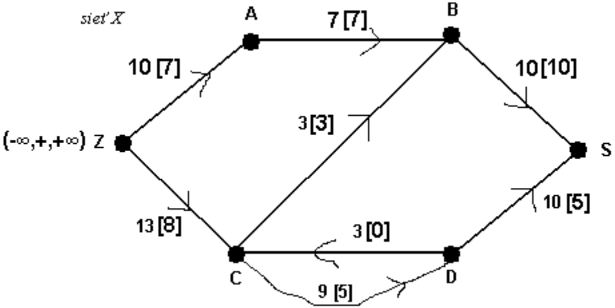
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



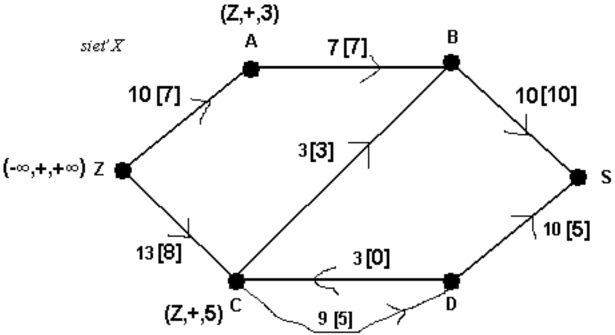
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



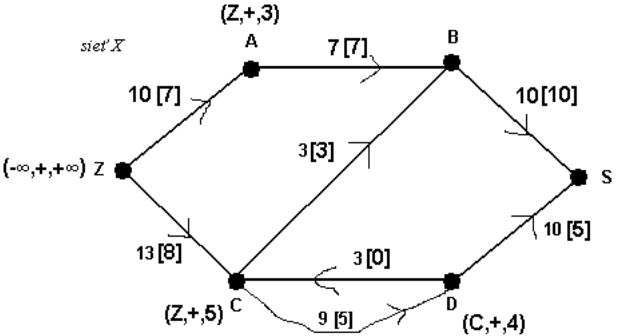
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



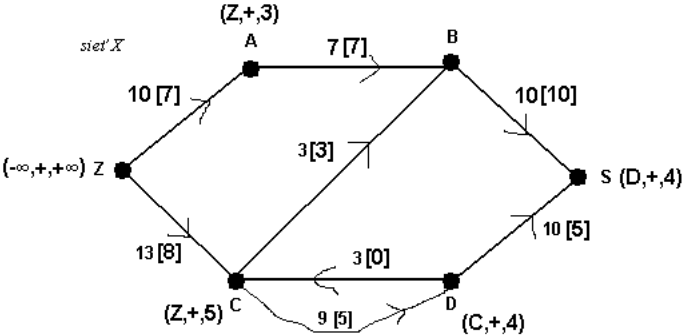
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



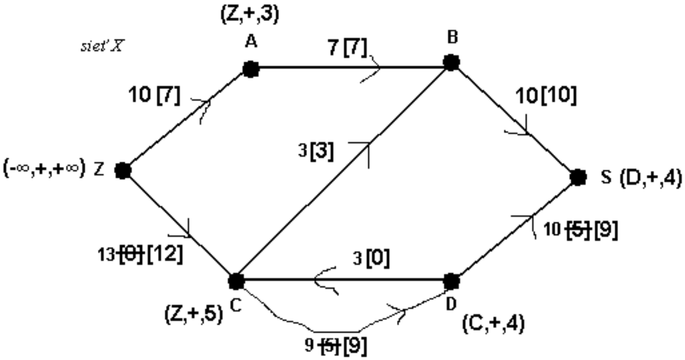
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



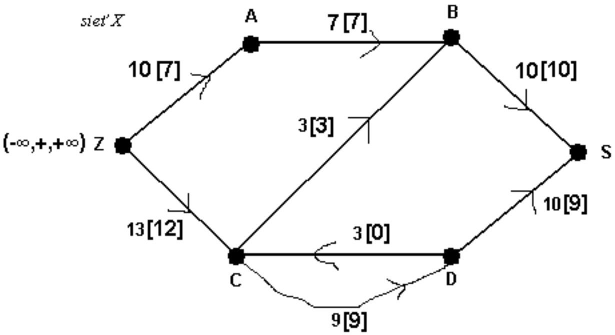
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



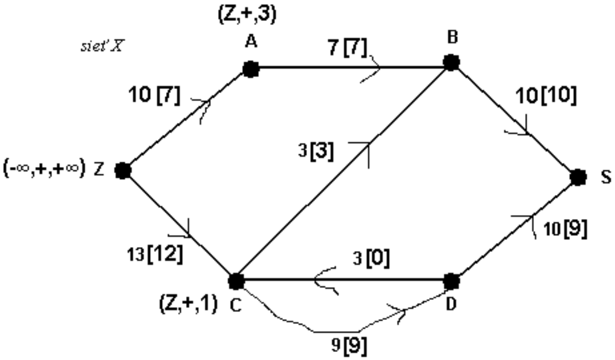
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



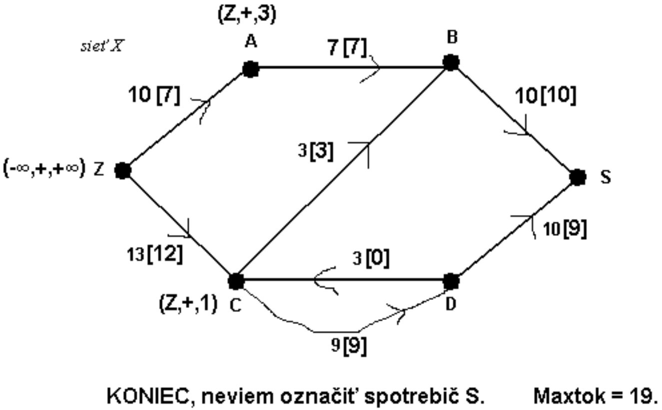
Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



Obrázok: Fordov-Fulkersonov algoritmus.

Príklad



Obrázok: Fordov-Fulkersonov algoritmus.

Rez, kapacita rezu

Definition

Nech $W \subseteq V(G), z \in W, s \in \overline{W} = V(G) - W$. Rez $(W, \overline{W}) = \{< u, v >, u \in W, v \in \overline{W}\}$. Kapacita rezu $|(W, \overline{W})| = c(W, \overline{W}) = \sum_{u \in W, v \in \overline{W}} c(< u, v >)$.

Netflow

Definition

Nech $U, V \subseteq V(X)$. $f(U, V) = \sum_{u \in U, v \in V} f(< u, v >)$.

$\text{Netflow}(U) = f(U, \overline{U}) - f(\overline{U}, U)$.

Netflow

Theorem

Nech f je tok v sieti X, a rez $(W, \overline{W})$ je taký, že $z \in W, s \in \overline{W} = V(G) - W$. Potom $|f| = netflow(W, \overline{W}) \leq c(W, \overline{W})$.

Bilancie zdroja a spotrebiča

Corollary

„Bilancia zdroja" = „Bilancia spotrebiča".

$$\sum_{Init(e)=z} f(e) - \sum_{Term(e)=z} f(e) = \sum_{Term(e)=s} f(e) - \sum_{Init(e)=s} f(e).$$

Tok v sieti a kapacita rezu

Corollary

Tok v sieti $\leq$ kapacita ľubovoľného rezu.

Zastavenie Fordovho-Fulkersonovho algoritmu

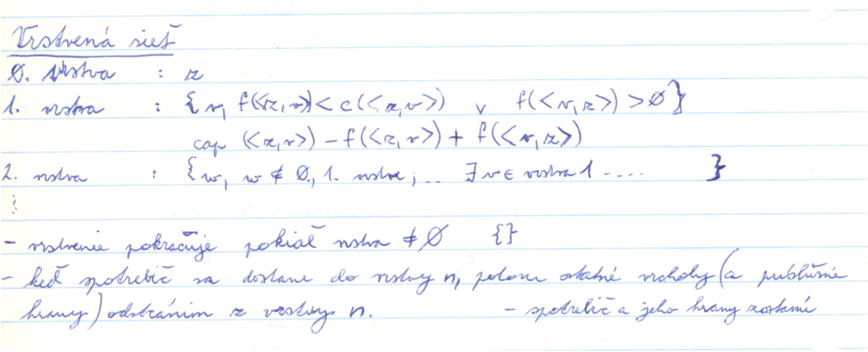
Corollary

Keď sa Fordov-Fulkersonov algoritmus zastaví s príčinou ,,tok je maximálny", potom tok v sieti je maximálny.

Zastavenie Fordovho-Fulkersonovho algoritmu

Theorem

V sieti s celočíselnými kapacitami a tokmi sa Fordov-Fulkersonov algoritmus zastaví po konečne veľa krokoch. Nájde hodnotu maximálneho toku v sieti.



Vrstvená sieť

0. vrstva : z

1. vrstva : $\{ v,\ f(\langle z,v\rangle) < c(\langle z,v\rangle) \vee f(\langle v,z\rangle) > 0 \}$

$cap(\langle z,v\rangle) - f(\langle z,v\rangle) + f(\langle v,z\rangle)$

2. vrstva : $\{ w,\ w \notin 0., 1.$ vrstva ; .. $\exists v \in$ vrstva 1 ... $\}$

⋮

- vrstvenie pokračuje pokiaľ vrstva $\neq \emptyset$ $\{\}$
- keď spotrebič sa dostane do vrstvy n, zahodím ostatné vrcholy (a príslušné hrany) odstránim z vrstvy n. – spotrebič a jeho hrany zostanú

Obrázok: Vrstvená sieť - princíp konštrukcie.

Vrstvená sieť - algoritmus konštrukcie

```
procedure vrstva (X, f, Y, maxflow);
  { vytvoríme vrstvenú sieť Y ku sieti X s tokom f.}
  { maxflow : logical ; = true ak tok je maximálny }
  L := Ø;
  vrstva (L) = {z};
  maxflow := false;
  repeat
     vrstva (L+1) := Ø;
     for u ∈ vrstva (L) do
        for (v nie je v žiadnej vrstve) alebo (vrstva (v) = L+1)
           q := c(<u,v>) - f(<u,v>) + f(<v,u>);
           if q > Ø then do begin
                 vrob hranu u -> v s kapacitou q;
                 daj v do vrstvy (L+1);
           end;
     L := L+1;
     if vrstva (L) = Ø then maxflow := true; exit;
  until spotrebič ∈ vrstva (L);
  vymaž z vrstvy L všetky iné vrcholy (než spotrebič) a príslušné hrany;
end;
```

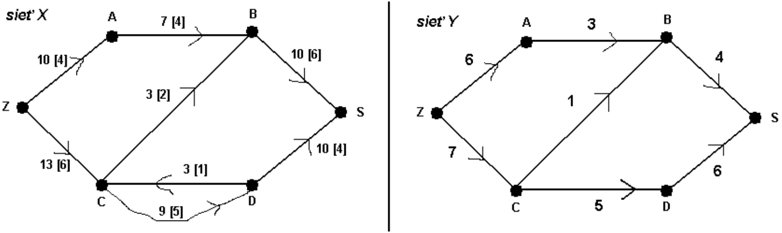
Obrázok: Vrstvená sieť - algoritmus konštrukcie.

Vrstvená sieť

Poznámky:

- tvorba vrstv. siete: zlúčenie reziduálnych kapacít
- vo vrstv. sieti sú všetky hrany orientované od zdroja k spotrebiču
- vo vrstv. sieti má každá cesta od zdroja k spotrebiču rovnaký počet hrán
- keď sa spotrebič dostane do poslednej vrstvy, to znamená, že sme našli zlepšujúcu cestu
- množiny vrcholov vrstvených sietí nemusia byť totožné

Príklad



Obrázok: Vrstvená sieť Y k pôvodnej sieti X.

Výška vrstvenej siete

Definition

Výška vrstvenej siete je dĺžka cesty od zdroja k spotrebiču.

Blokovací tok

Definition

Tok je blokovací práve vtedy, keď má každá cesta od zdroja k spotrebiču aspoň jednu nasýtenú hranu.

Dinicov algoritmus

1. Začneme v nulovým tokom f v sieti X.
2. K sieti X vytvoríme vrstvenú sieť Y. Ak to nie je možné, tak koniec, bol nájdený maximálny tok.
3. Vo vrstvenej sieti Y nájdeme blokujúci tok.
4. Upravíme tok v pôvodnej sieti X a prejdeme na krok 2.

Úprava toku v pôvodnej sieti

```
procedure zväčšenie (f, X, Y, g);
  zväčši tok f v X pomocou (použitím) blokujúceho toku g v Y},
  for každá hrana e: u -> v ∈ Y do
  begin
                        (v smere X)
    zväčši tok f: u -> v o hodnotu
      min {g(e), c(<u,v>) - f(<u,v>)};
    if neminuli sme celé g(<u,v>) then
        zmenši tok f(<v,u>) o zvyšok g(e).
  end;
end;
```

Obrázok: Úprava toku v pôvodnej sieti.

Zastavenie Dinicovho algoritmu

Theorem

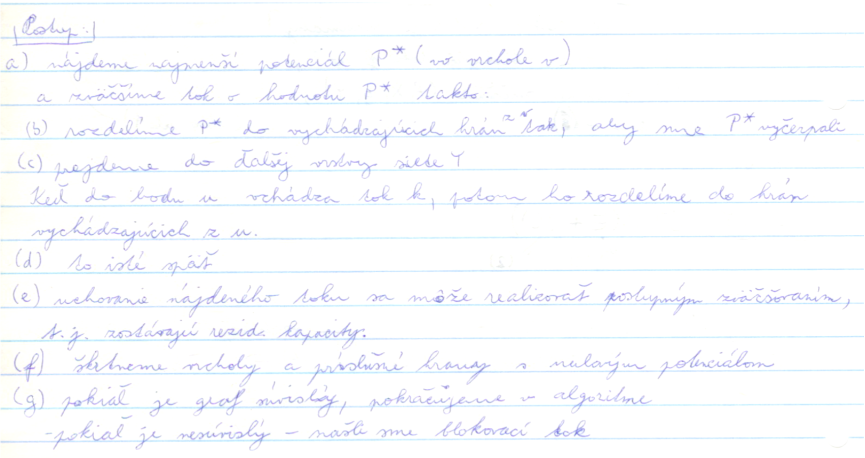
Keď v $Y(p+1)$ *vrstvenej sieti máme cestu* $v_0, v_1, v_2, \ldots, v_a$ *takú, že* $v_0, v_1, v_2, \ldots, v_a \in Y(p)$*, potom*
$v_a \in vrstva(b) \in Y(p) \rightarrow a \geq b.$

Zastavenie Dinicovho algoritmu

Theorem

Výšky vrstvených sietí ku X tvoria rastúcu postupnosť.

Nájdenie blokujúceho toku vo vrstvenej sieti

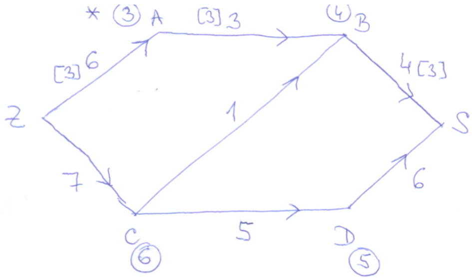


Postup:

a) nájdeme najmenší potenciál P^* (vo vrchole v)
a zväčšíme tok o hodnotu P^* takto:
(b) rozdelíme P^* do vychádzajúcich hrán z v tak, aby sme P^* vyčerpali
(c) prejdeme do ďalšej vrstvy siete T
Keď do bodu u vchádza tok k, potom ho rozdelíme do hrán vychádzajúcich z u.
(d) to isté späť
(e) uchovanie nájdeného toku sa môže realizovať postupným zväčšovaním, t.j. zostávajú rezid. kapacity.
(f) škrtneme vrcholy a príslušné hrany s nulovým potenciálom
(g) pokiaľ je graf súvislý, pokračujeme v algoritme
-pokiaľ je nesúvislý - našli sme blokovací tok

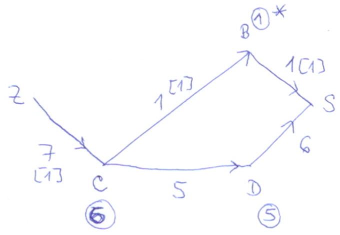
Obrázok: MPM algoritmus na nájdenie blokujúceho toku.

Príklad



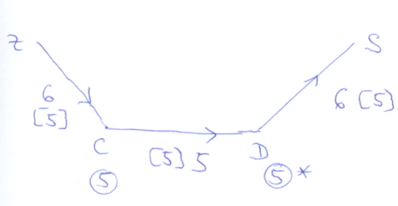
Obrázok: Nájdenie max.toku pomocou vrstvených sietí.

Príklad



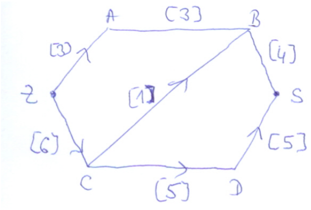
Obrázok: Nájdenie max.toku pomocou vrstvených sietí.

Príklad



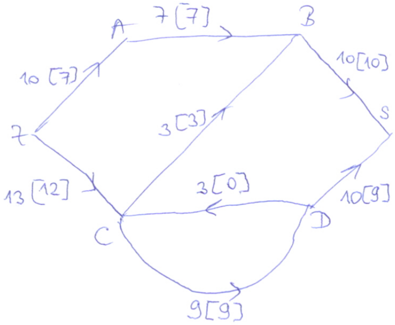
Obrázok: Nájdenie max.toku pomocou vrstvených sietí.

Príklad



Obrázok: Nájdenie max.toku pomocou vrstvených sietí.

Príklad



Obrázok: Nájdenie max.toku pomocou vrstvených sietí.

Príklad

Obrázok: Nájdenie max.toku pomocou vrstvených sietí.

Aplikácie

- toky v rozvodných sústavách
- párovanie v grafe
- hranová súvislosť grafu