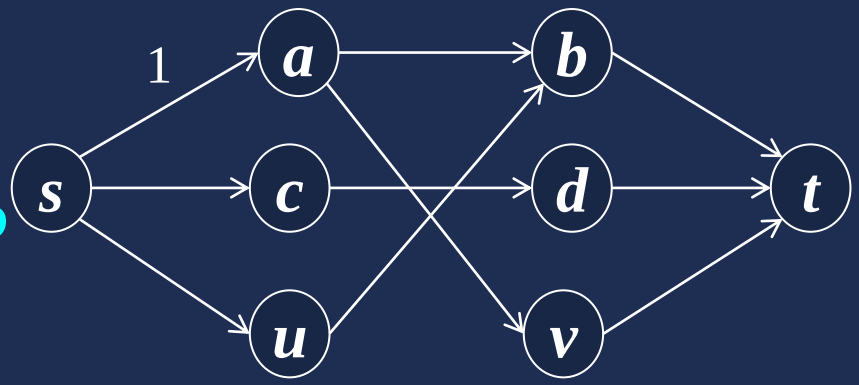
网络流

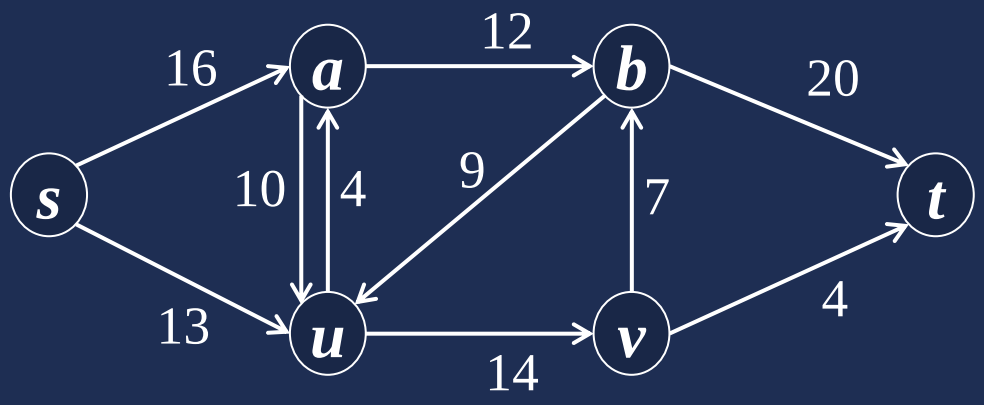
流网络定义

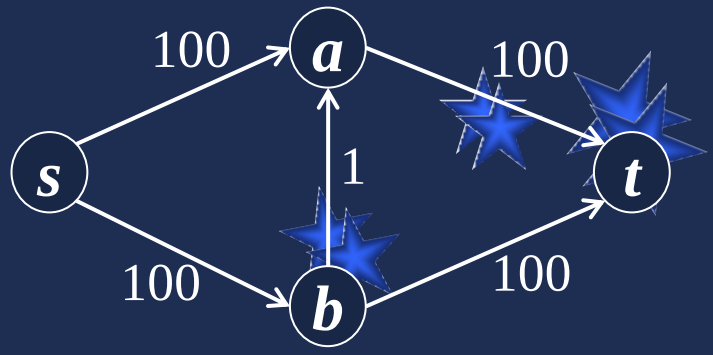
- 有向图G=(V, E) 中： (假设不存在平行边)
  - 有唯一的一个源点 s （入度=0）
  - 有唯一的一个汇点 t （出度=0）
  - 图中每条弧 (u, v) 都有一个非负容量 c(u, v)
- 满足上述条件的图G称为网络流图
- 记为： G=(V， E ， C， s ， t)

最大流

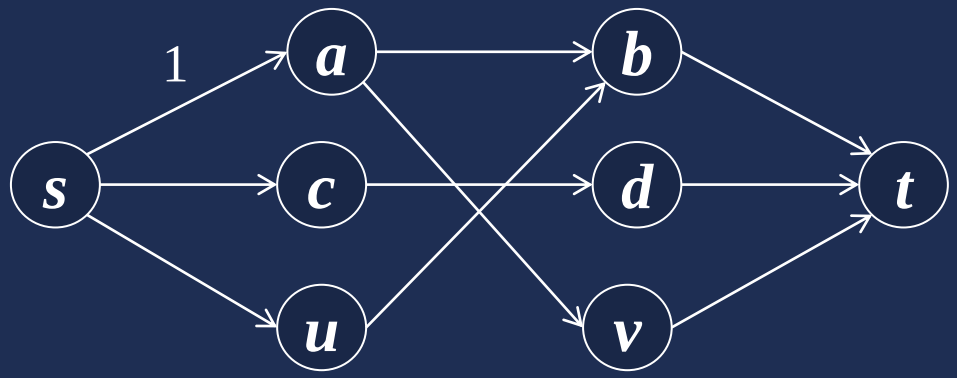
- 有向图G=(V,E), 容量c: V×V→R≥0, 源s, 汇t,
- 求容量限制下, 能从s运送到t的最大流量
- c(s,a)=16, c(b,v)=0, ... //只画容量非零边
- 如何增加流量?
- 如何判断已达最大量?

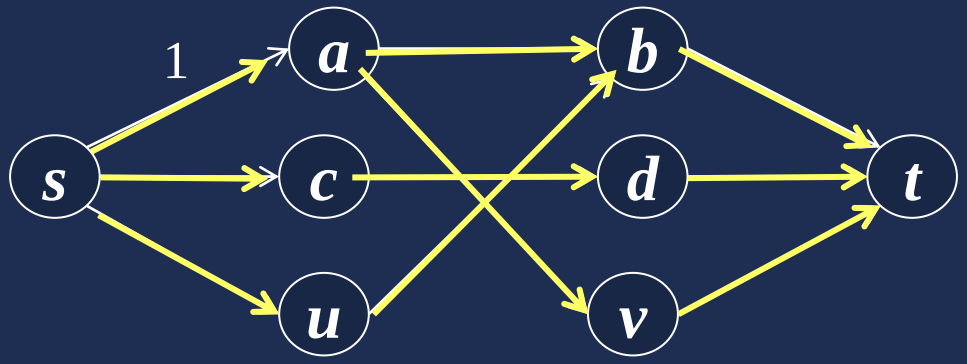






如何增加流量





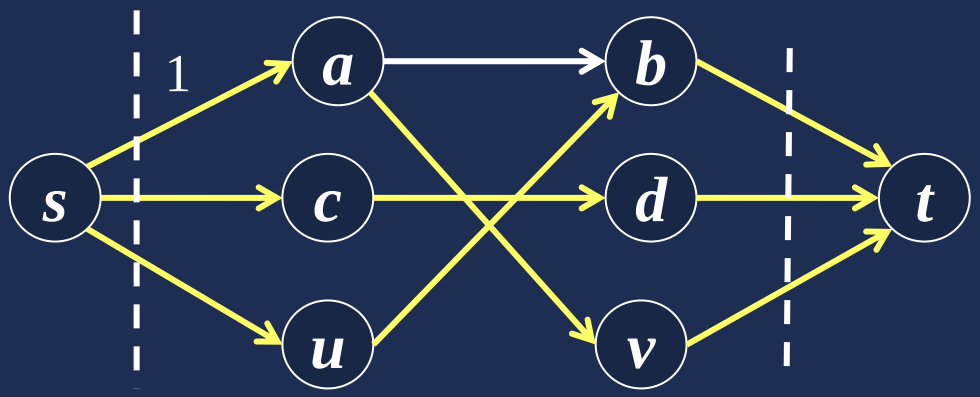
s→a→b→t 1

s→c→d→t 1

s→u→b→a→v→t 1

如何实现回退?

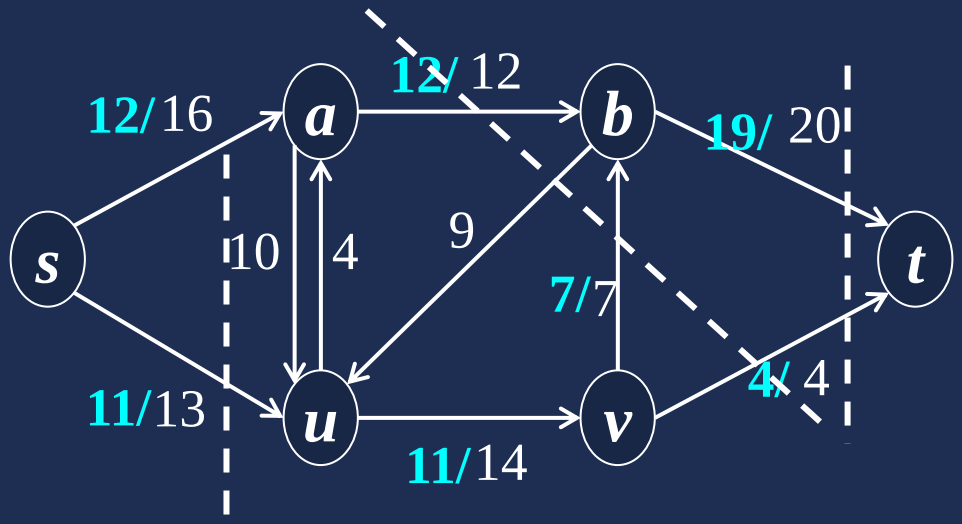
如何判断不能再增大流量



流量下界3

流量上界4, 3

最大流=3



s→a→b→t 12

s→u→v→b→t 7

s→u→v→t 4

流量下界23

流量上界29, 24, 23

最大流 = 23

最大流最小割定理: 一定存在这样的上界

流，流量，最大流

- 对于流网络 G（假设不存在平行边和圈）
- 称 f: V×V→R为可行流, 若f 满足:

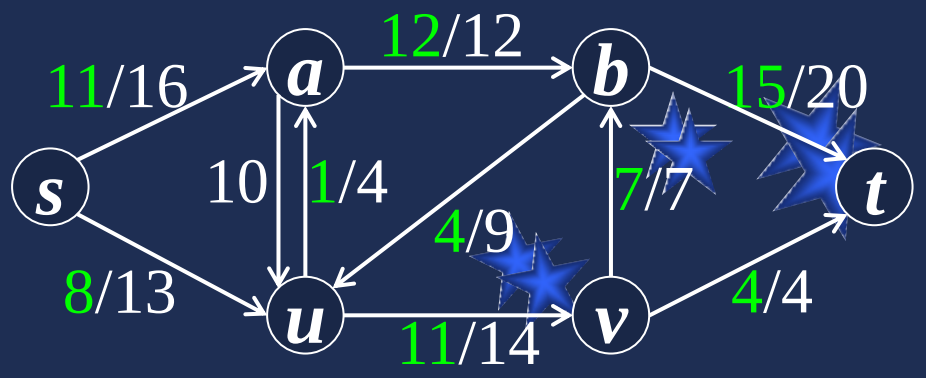
(1) 容量限制, $0 \le f(u,v) \le c(u,v)$　//图中只画非0边

(2) 流守恒性, $\forall u \neq s,t,\ \Sigma_{v \in V} f(u,v) = \Sigma_{v \in V} f(v,u)$

- s的净流出量 $|f| = \Sigma_{v \in V} f(s,v) - \Sigma_{v \in V} f(v,s)$
- 最大流: 流网络上的流f 满足

$|f| = \max\{ |h| : h$是G的流 $\}$

- 注: f(u,v),f(v,u)不同时>0.
- 若(u,v)∉E, 则f(u,v)=0.



定义: 割(S,T), 割的净流

对于流网络G, s, t, c 和流 f

$\forall A,B \subseteq V,\ f(A,B) = \sum_{u\in A, v\in B} f(u,v) - \sum_{u\in A, v\in B} f(v,u)$

$f(A, B) = -f(B, A),\quad f(A, A) = f(B, B) = 0$

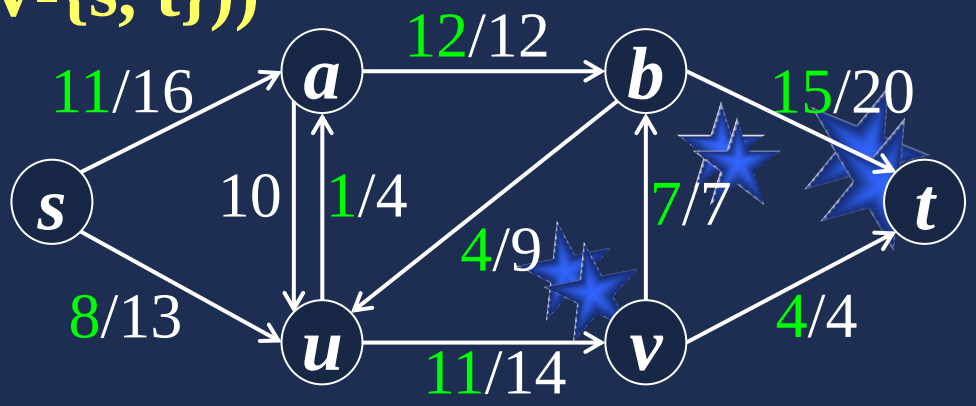
- 性质: $\forall u \neq s,t,\ f(u,V) = f(V,u) = 0$. //f(s,V)=19
- 定义: 割(S,T)满足 (1) T=V-S (2) $s\in S,\ t\in T$.
- 性质: $|f| = f(s,V) = f(V,t)$ //s的净流出量=t的净流入量

证明: $f(V,t) = f(V, V - \{s\} - (V-\{s, t\}))$

$= f(V,V) - f(V,s) - f(V,V-\{s,t\})$

$= -f(V,s)$

$= f(s,V)$



定义: 割(S,T), 割的净流

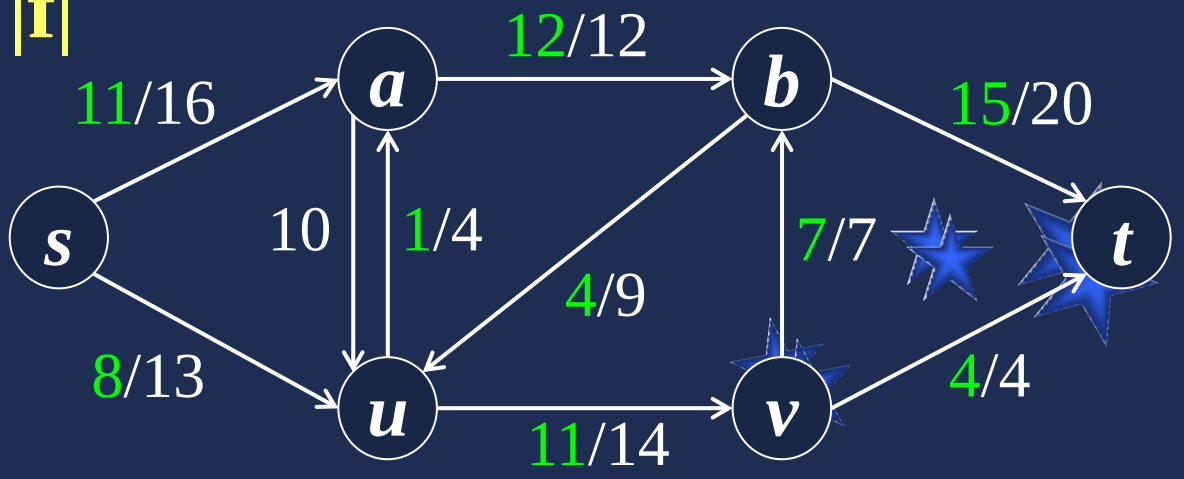
对于流网络G, s, t, c 和流 f

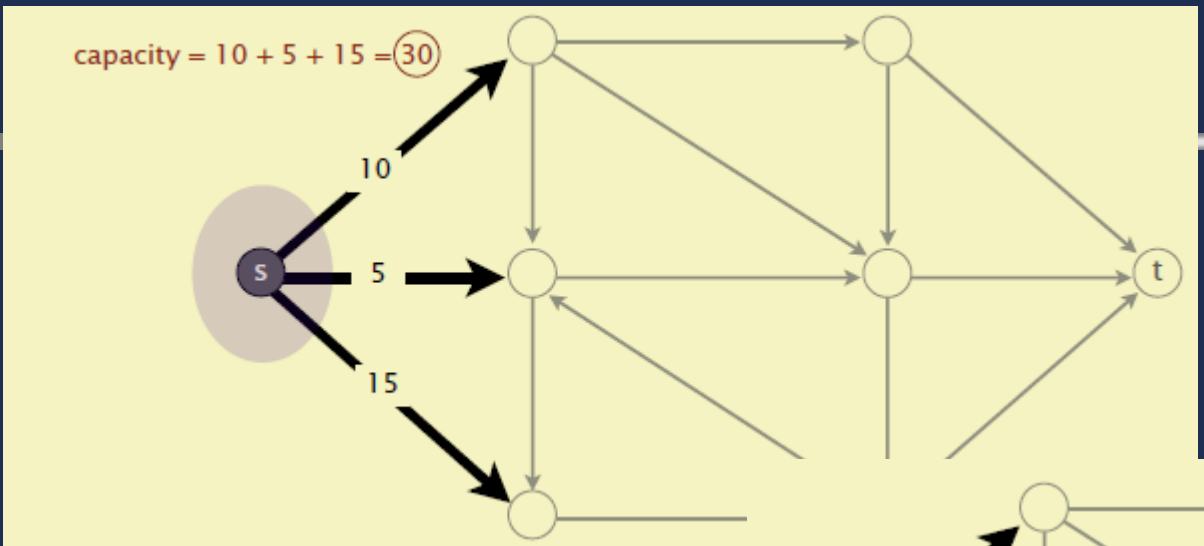
- 定义: 割(S,T)满足 (1) T=V-S (2) s∈S, t∈T.
- 性质: |f| = f(S,T) //割的净流

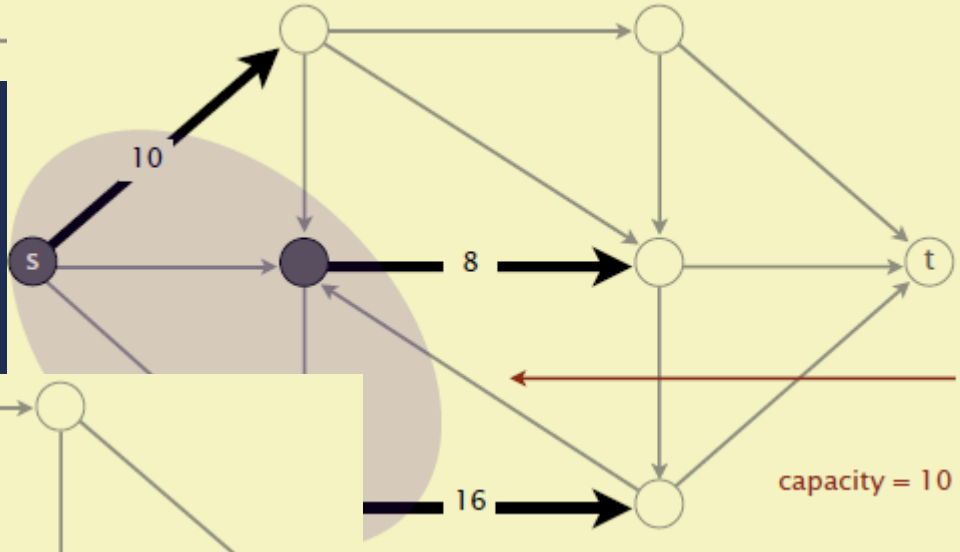
证明: f(S,T) = f(S,V) – f(S,S)

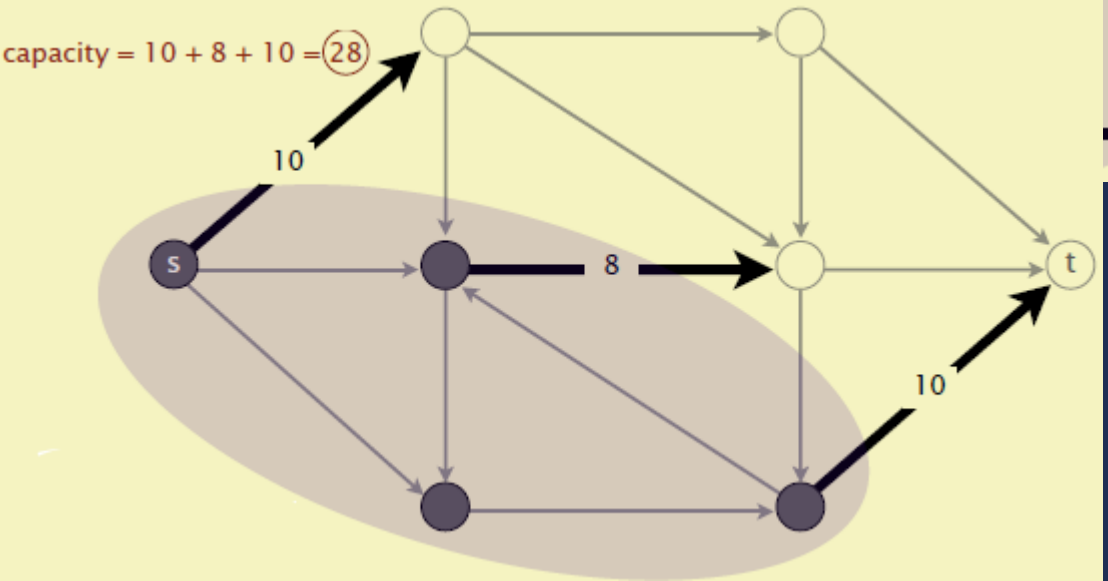
= f(s,V) + f(S-{s}, V)

= f(s,V) = |f|



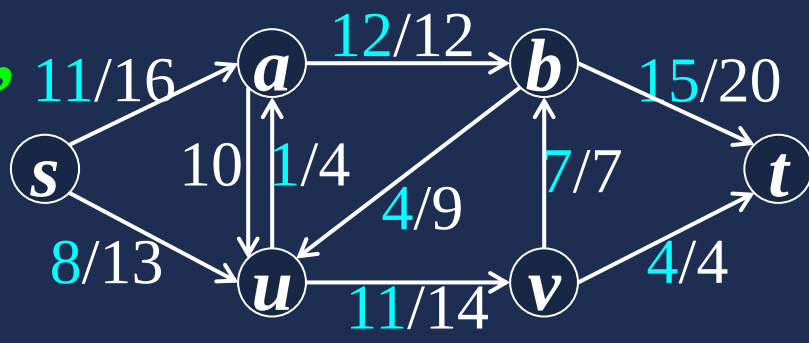

capacity = 10 + 5 + 15 = 30
10
5
15
s
t


10
8
16
s
t
don't count edge
from B to A
capacity = 10 + 8 + 16 = 34


capacity = 10 + 8 + 10 = 28
10
8
10
s
t

最大流与最小割

- 割(S,T): (1) T=V-S (2) s∈S, t∈T.
- 性质: $|f|=f(S,T)=\Sigma_{u\in S,v\in T}f(u,v)-\Sigma_{u\in S,v\in T}f(v,u)$
- 定义: 割(S,T)的容量: $c(S,T)=\Sigma_{u\in S,v\in T}c(u,v)$
- 性质: ∀割(S,T), ∀流f, $f(S,T)\le c(S,T)$
- 推论: 若流f满足, $|f|=c(S,T)$, 则f是最大流, (S,T)是最小割

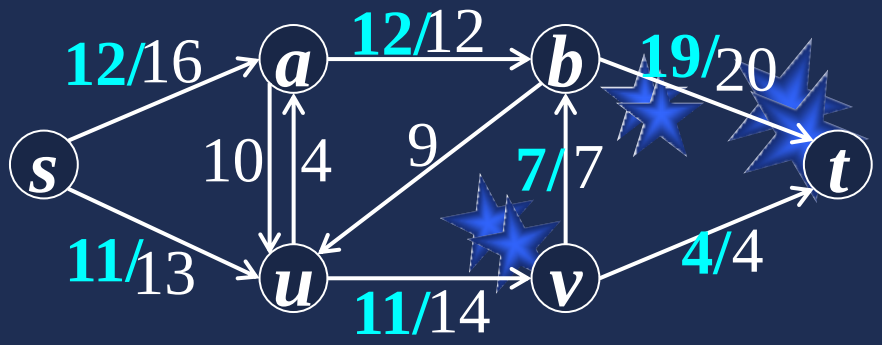


c({s}, {a,b,u,v,t})=29

c({s,a,u}, {b,v,t})=26

c({s,a,b,u,v}, {t})=24

c({s,a,u,v}, {b,t})=23



剩余网络

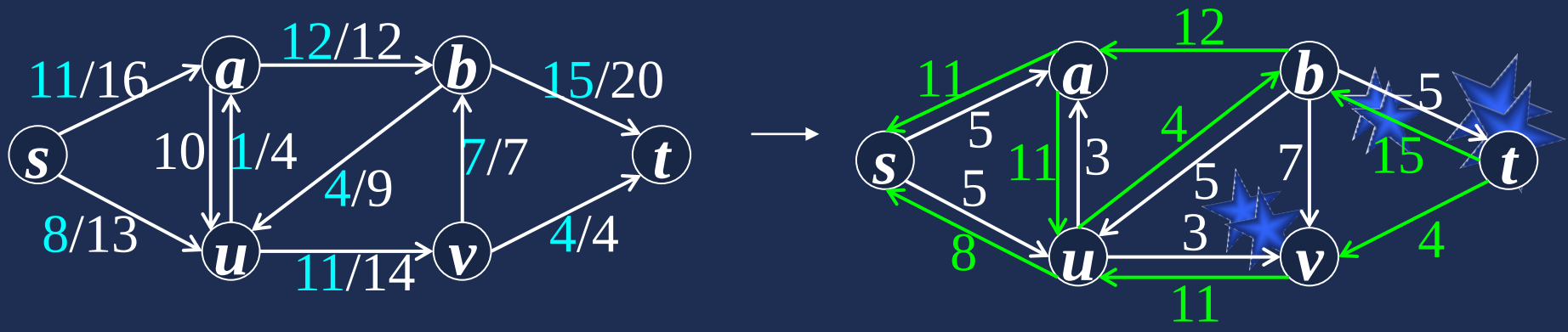
给定流网络G, s, t, c 和流f

- 流f的剩余容量, $c_f(u,v) = c(u,v) - f(u,v)$, 若$f(u,v)>0$;
  $= c(u,v) + f(v,u)$, 若$f(v,u)>0$;
  $= c(u,v)$, 否则.

求$c_f(s,a)$, $c_f(a,s)$, $c_f(a,b)$, $c_f(b,a)$,

5 11 0 12

- 剩余网络, $G_f = (V, E_f)$, $E_f = \{ (u,v) \mid c_f(u,v)>0 \}$



剩余网络

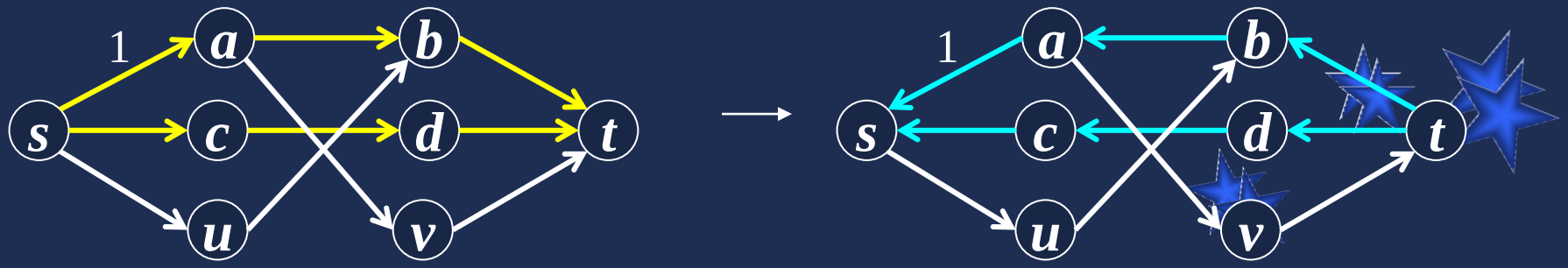
给定流网络G, s, t, c 和流f

- 流f的剩余容量, $c_f(u,v) = c(u,v) - f(u,v)$, 若$f(u,v)>0$;
  $= c(u,v) + f(v,u)$, 若$f(v,u)>0$;
  $= c(u,v)$, 否则.

求$c_f(s,a)$, $c_f(a,s)$, $c_f(a,b)$, $c_f(b,a)$,

0 1 0 1

- 剩余网络, $G_f = (V, E_f)$, $E_f = \{ (u,v) \mid c_f(u,v)>0 \}$



剩余网络的作用:找增广路径

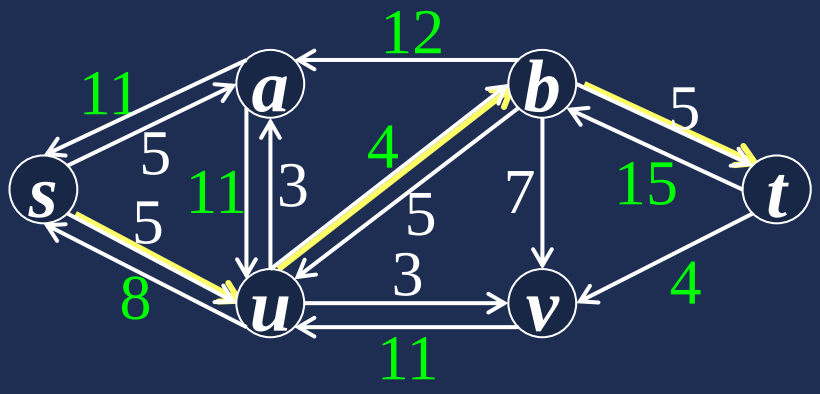
- 定义: 增广路径是G_f中从s到t的一条简单路径
- 定义: 若p为增广路径, 则称

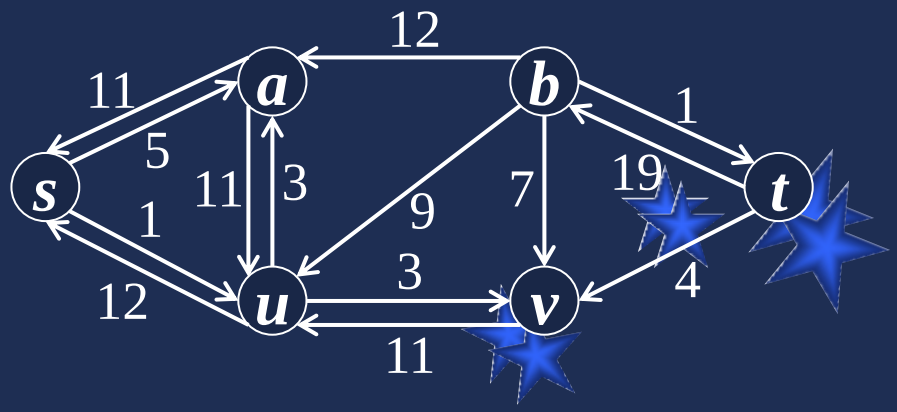
$$c_f(p) = \min \{ c_f(u,v) : (u,v) \in p \}$$

为p的剩余容量, 是G_f中能沿p传输的最大量.

注: 有增广路径 ⇒ (1)可以增大流量 (2) f不是最大流

$$c_f(p) > 0$$





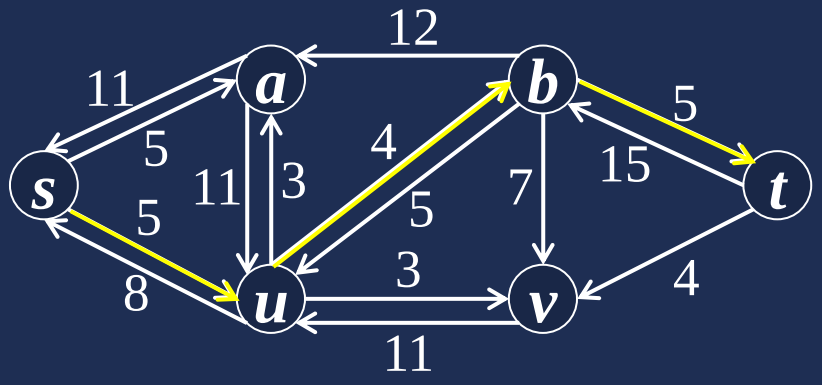
增广流

给定G, s, t, c, f, G_f 和增广路径p,

- 定义沿p的增广流f_p: $\forall u,v \in V$,

  $f_p(u,v) = c_f(p)$, 若(u,v)在p上;

  $= 0$, 否则.

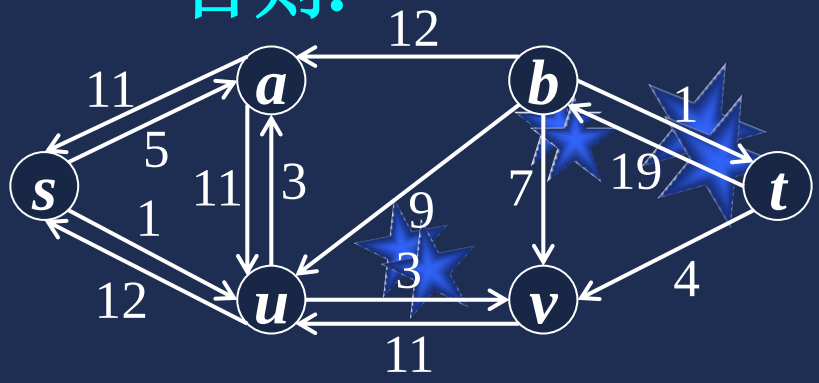


- 注: $f_p(u,v)$, $f_p(v,u)$不同时大于0.

- f_p是剩余网络G_f上的流, $|f_p| = c_f(p)$.
- $(f + f_p)(u,v) = f(u,v) + f_p(u,v) - f_p(v,u)$, 若$(u,v) \in E$;

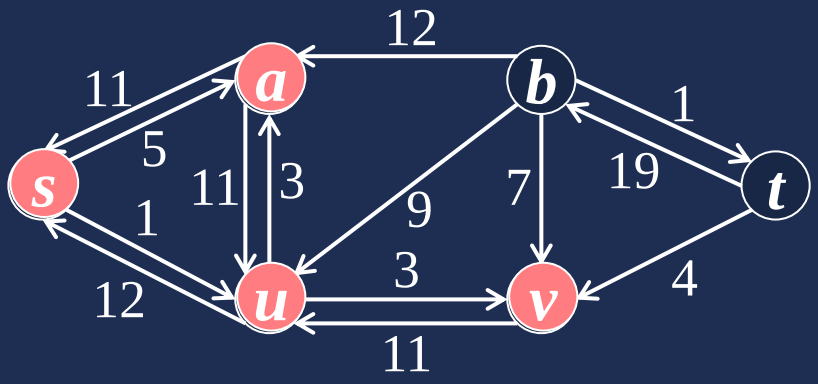
  $= 0$, 否则.
- $f + f_p$是流网络G上的流且

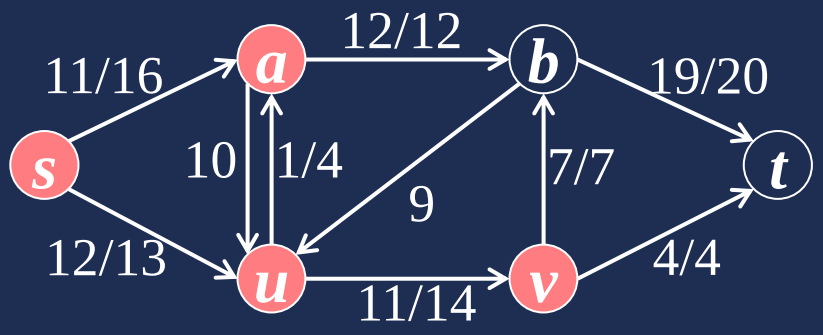
  $|f + f_p| = |f| + c_f(p) > |f|$.
- 如果没有增广路径怎么办?



无增广路径则得最小割

- 若有 |f| = c(S,T), 则f是最大流, (S,T)是最小割
- 如何判断G_f中无增广路径? s,t不连通





性质: 设G_f无增广路径, 取S={G_f中s能到达的点},T=V-S,
则|f| = f(S,T) = c(S,T), 即f是最大流, (S,T)是最小割.

证明: (1) t∈T, T≠∅.

(2)∀u∈S, v∈T, $c_f(u,v)=0$ (否则v∈S)

(3) f(S,T) = c(S,T). ∀u∈S, v∈T, $f(v,u)>0 \Rightarrow c_f(u,v)=c(u,v)+f(v,u)>0$,矛盾;

$f(u,v)>0 \Rightarrow c_f(u,v)=c(u,v)-f(u,v)=0$; 否则,

$c_f(u,v)=c(u,v)=f(u,v)=f(v,u)=0$

//$c(S,T) = \Sigma_{u\in S,v\in T}c(u,v)$, $f(S,T) = \Sigma_{u\in S,v\in T}f(u,v) - \Sigma_{u\in S,v\in T}f(v,u)$

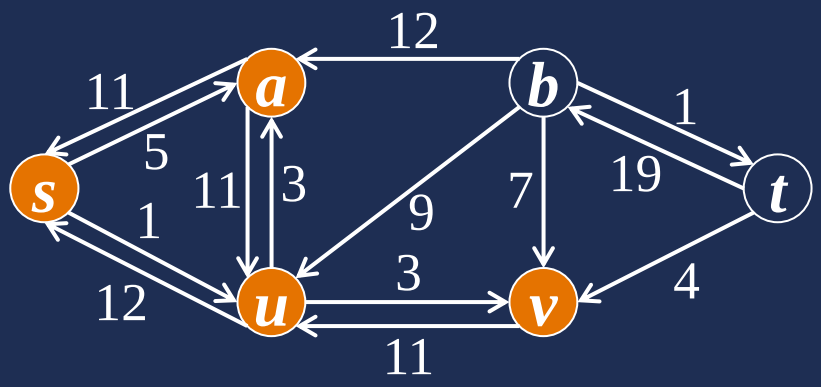
最大流最小割定理

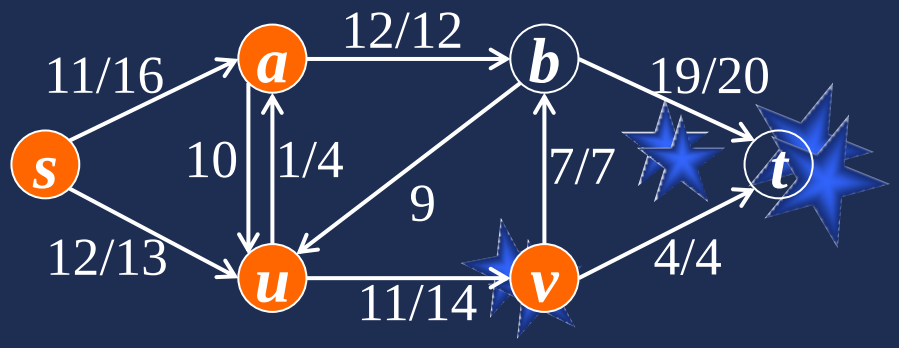
- 性质: ∀流f, ∀割(S,T), $|f| \le c(S,T)$.
- 定理(最大流最小割) 下列条件等价

  (1) f是G的一个最大流;

  (2) G_f不包含增广路径;

  (3) 存在割(S,T)使得 $|f| = c(S,T)$.
- 证明: (1)⇒(2), (2)⇒(3), (3)⇒(1)



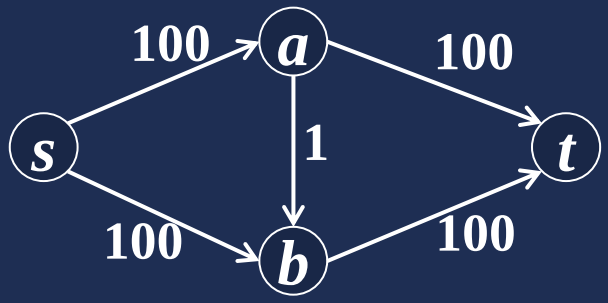


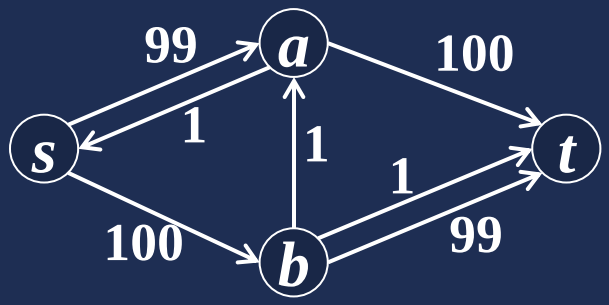
Ford-Fulkerson算法

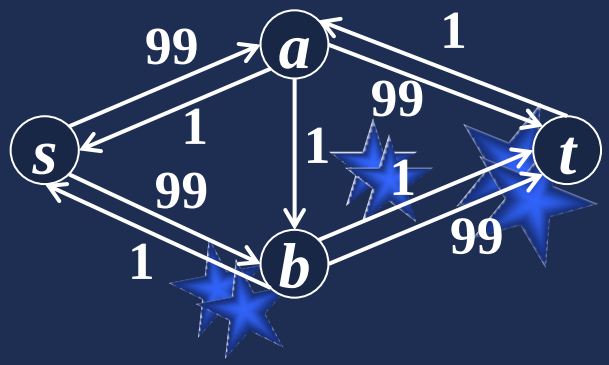
1. 从零流开始
2. 当剩余网络中有增广路径时,
3. 找一条增广路径, 增大流,
4. 输出流量

- 当容量是整数时, 时间复杂度O(|E| f*)

  f*是最大流流量







算法复杂度

Ford-Fulkerson, 容量为整数, $O(|E|\ |f^*|)$

Edmond-Karp, $O(|V|\ |E|^2)$

一般push-relabel, $O(|V|^2\ |E|)$

SAP, $O(|V|^2\ |E|)$

Dinic, $O(|V|^2\ |E|)$

Relabel to front $O(|V|^3)$

Edmond-Karp算法([C])

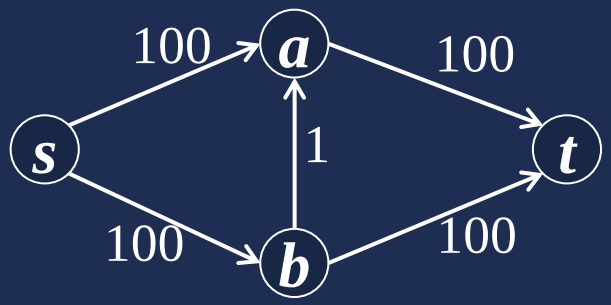
1. 从零流开始

2. 当剩余网络中有增广路径,

3. 找最短增广路径, 增大流,

4. 输出当前流量

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



Guardian of Decency

一保守教师想带学生郊游, 却怕他们途中谈恋爱,

他认为满足下面条件之一的两人谈恋爱几率很小:

(1) 身高差>40　　(2) 性别相同

(3) 爱好不同类型的音乐　(4) 爱好同类型的运动

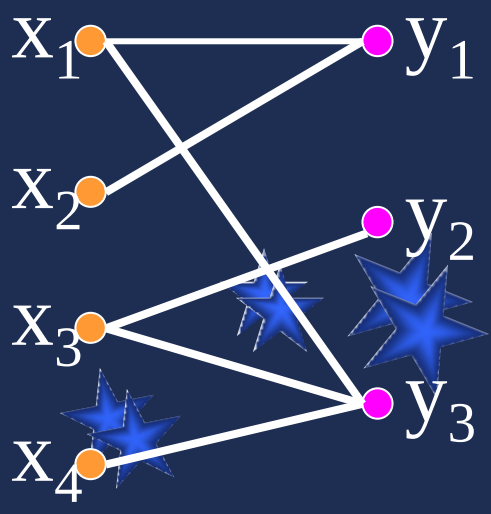
输入是学生的数据, 求最多能带多少学生. 例:

35 M classicism programming

0 M baroque skiing

43 M baroque chess

30 F baroque soccer



二分图匹配

定义: 对图G=(V,E), 称M⊆E为匹配,

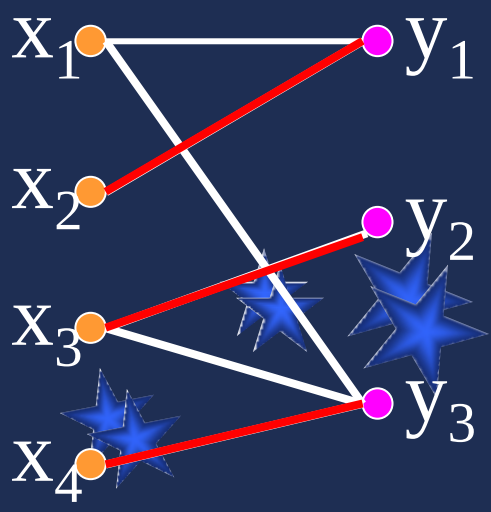
若M中任意两条边没有公共顶点.

G的最大匹配数定义为

$\rho(G) = \max\{ |M| \mid M是G的匹配 \}$

性质: 对二分图G=(X,E,Y), $\rho(G) \le \min\{|X|,|Y|\}$

定义: 满足 $|M^*| = \rho(G)$ 的匹配M^*
称为最大匹配.



最小覆盖

定义: 对图G=(V,E), 称S⊆V为覆盖,

若G中任意边都有顶点在S中.

G的最小覆盖数定义为

$c(G)=\min\{\ |S|\ |\ S是G的覆盖\ \}$

引理: G是图, 则 $\rho(G) \le c(G)$.

König定理: G是二分图, 则 $\rho(G) = c(G)$.

注: König定理对一般图不成立. 举例?

一般图找最小覆盖是NP完全问题

一般图找最大匹配有多项式时间的开花算法

最大独立集

定义: 对图G=(V,E), 称T⊆V为独立集,

若T中任两顶点都无G中的边相连.

G的最大独立集定义为

$\alpha(G) = \max\{ |S| \mid S是G的独立集 \}$

引理: S是G的覆盖⇔V-S是G的独立集.

推论: $\alpha(G) = |V| - c(G)$.

保守教师解法

将男女生分为左右顶点集,

若有男女生满足

身高差≤40, 音乐爱好相同, 运动爱好不同,

则添边.

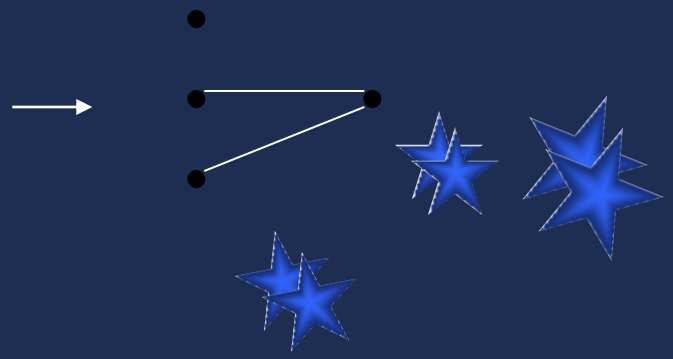
保守教师带的人是此二分图的最大独立集.

35 M classicism programming

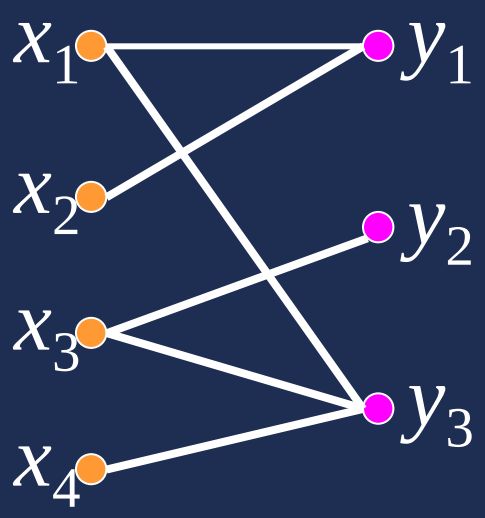
0 M baroque skiing

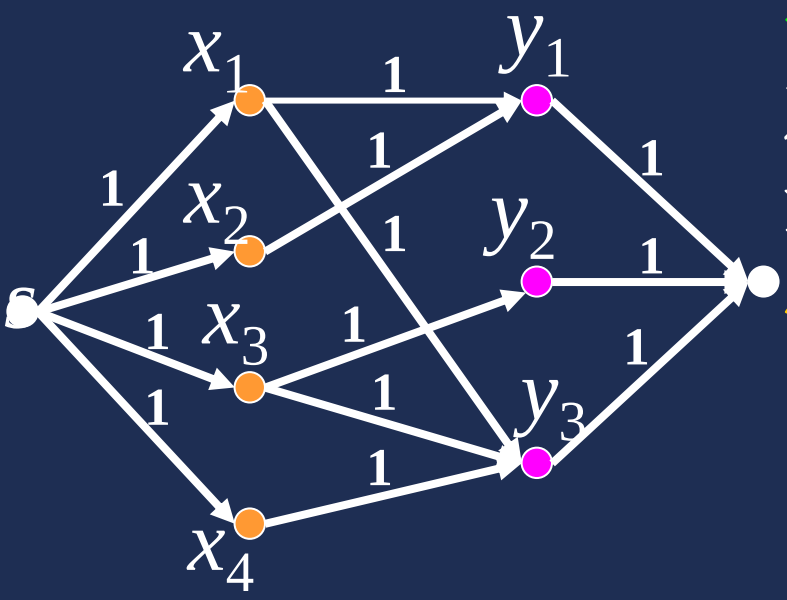
43 M baroque chess

30 F baroque soccer



将二分图匹配转化为最大流





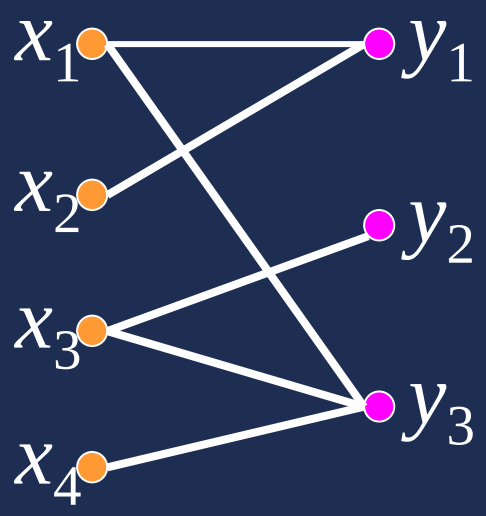
男生1:k对应的顶点x[1:k]，女生1:n对应的顶点y[1:n]，添加源点s,和汇点t.

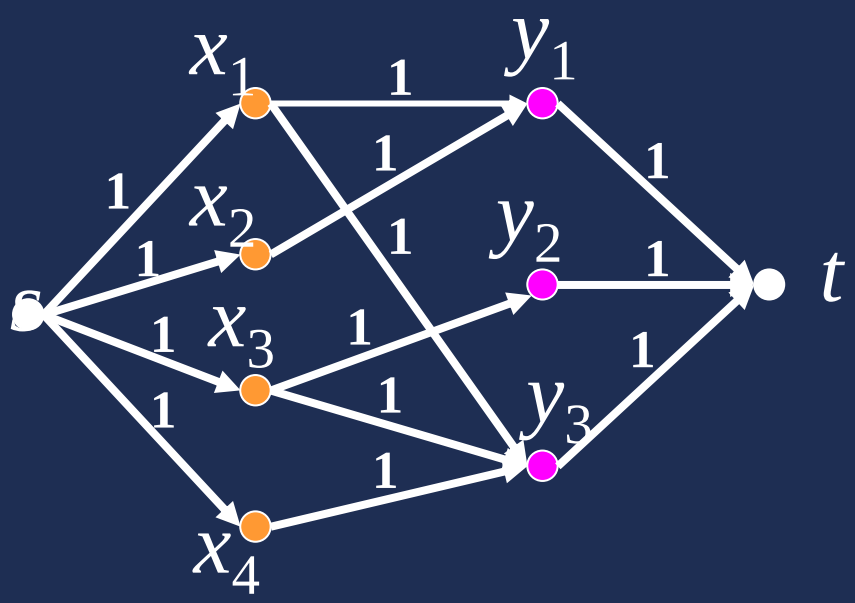
从s各连1条边到k个顶点x[1:k], 容量1， 若身高差≤40, 音乐爱好相同, 运动爱好不同, 则添边(x[j],y[i]), 容量1

对每个女生i, 添加1条边(y[i],t), 容量1

使用最大流算法, 得到相应解.

将二分图匹配转化为最大流





定理: 将二分图G_1如上转化为流网络G_2,s,t,c

则$\rho(G_1)$等于G_2的最大流量.

证明: (1) 存在整数最大流f: $V\times V\rightarrow\{1,0,-1\}$;

(2) 整数流→匹配

(3) 匹配→流. 所以 $\rho(G_1) = |f|$.

O(|V||E|)

本章作业

1. 飞行员配对

问题描述: 第二次世界大战时期, 英国皇家空军从沦陷国征募了大量外籍飞行员. 由皇家空军派出的每一架飞机都需要配备在航行技能和语言上能互相配合的2名飞行员, 其中一名是英国飞行员, 另一名是外籍飞行员, 在众多的飞行员中, 每一名外籍飞行员都可以与其他若干名英国飞行员很好地配合. 如何选择配对的飞行员才能使一次派出最多的飞机.

算法设计: 对于给定的外籍飞行员与英国飞行员的配合情况, 找出一个最佳飞行员配对方案, 使得皇家空军能派出最多的飞行员.

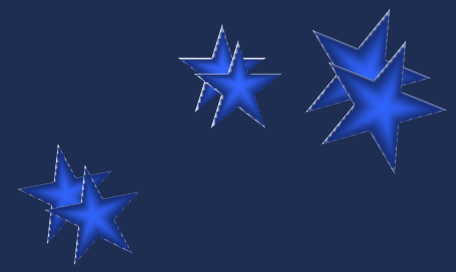
数据输入: 由文件input.txt提供输入数据. 文件第1行有2个正整数m和n. n是皇家空军的飞行员总数(n<100); m是外籍飞行员数. 外籍飞行员编号1~m, 英国飞行员编号m+1~n. 接下来每行2个整数i和j, 表示外籍飞行员i可以与英国飞行员j配合. 文件最后以2个-1结束.

本章作业

结果输出: 将最佳飞行员配对方案输出到文件output.txt. 第1行是最佳飞行员配对方案一次能派出的最多飞机数M. 接下来M行是最佳飞行员配对方案. 每行有2个正整数i和j, 表示在最佳飞行员配对中, 飞行员i和飞行员j配对.

如果所求的最佳飞行员配对方案不存在, 则输出“No Solution!”

输入文件示例	输出文件示例
5 10	Output.txt
1 7	4
1 8	1 7
2 6	2 9
2 9	3 8
2 10	5 10



本章作业

2. 试题库问题

问题描述: 假设一个试题库中有n道试题. 每道试题都标明了所属类别. 同一道题可能有多个类别属性. 现要从题库中抽取m道题组成试卷. 并要求试卷包含指定类型的试题. 试设计一个满足要求的组卷算法.

算法设计: 对于给定的组卷要求, 计算满足要求的组卷方案.

数据输入: 由文件input.txt提供输入数据. 文件第1行有2个正整数k和n(2≤k≤20, k≤n≤1000), k表示题库中试题类型总数, n表示题库中试题总数. 第2行有k个正整数, 第i个正整数表示要选出的类型i的题数. 这k个数相加就是要选出的总题数. 接下来n行给出了题库中每个试题的类型信息. 每行的第1个正整数p标明该题可以属于p类, 接着的p个数是该题所属的类型号.

结果输出: 将组卷方案输出到文件output.txt. 文件第i行输出“i:”后接类型i的题号. 如果有多个满足要求的方案, 只要输出1个方案

本章作业

输入文件示例

```
3 15
3 3 4
2 1 2
1 3
1 3
1 3
1 3
3 1 2 3
2 2 3
2 1 3
1 2
1 2
2 1 2
2 1 3
```

输出文件示例

```
Output.txt
1: 1 6 8
2: 7 9 10
3: 2 3 4 5
```

解: 构造流网络. 比如最大流量就是最终解.

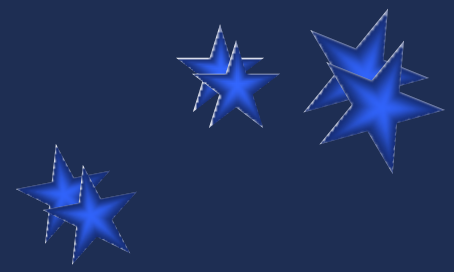
顶点构造:

...

边的构造:

...

使用最大流算法, 得到相应解.



附录

Edmond-Karp算法$O(|V||E|^2)$[C]

- 算法过程: 找最短增广路径, 增大流
- 性质: 流f, 剩余网络G_f, $\forall u$, 最短距离 $\delta_f(s,u) \le |V|$.
- 性质: 设G_f经一次增广得到$G_{f'}$, 则$\forall u$, $\delta_f(s,u) \le \delta_{f'}(s,u)$

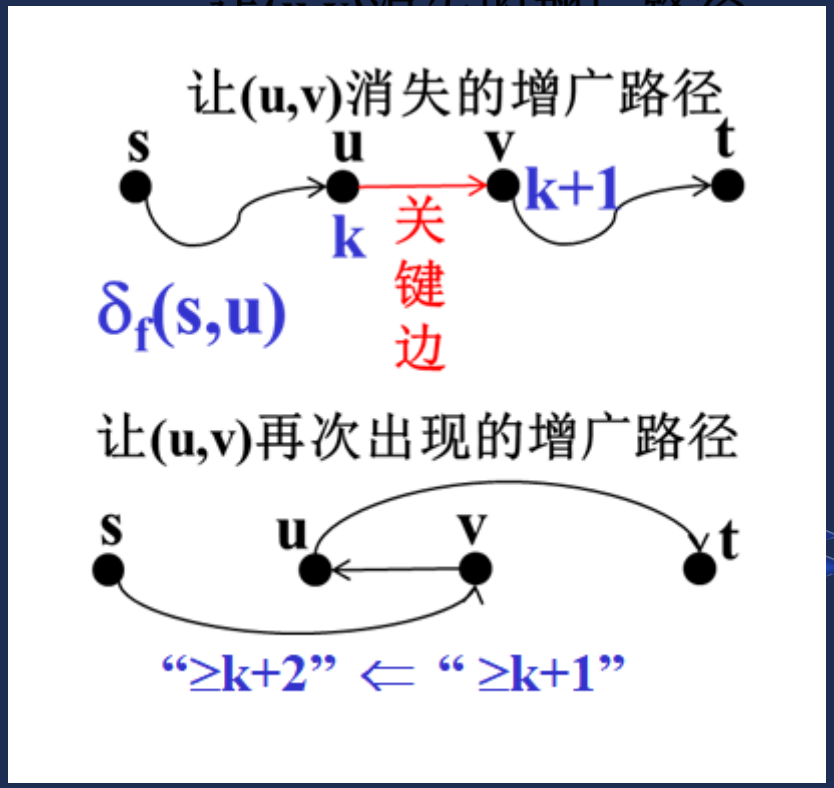
- 每次增广至少一条边消失(关键边)
- (u,v)消失后再次出现

⇔ (v,u)在当前增广路径上

⇒ v到s的距离至少增加2

⇒ (u,v)至多消失|V|/2次

- 共2|E|条边? 增广最多|V||E|次?
- 增广1次O(|E|)时间?



Edmond-Karp算法$O(|V||E|^2)$

- 算法过程: 找最短增广路径, 增大流
- 性质: 流f, 剩余网络G_f, $\forall u$, 最短距离 $\delta_f(s,u) \le |V|$.
- 性质: 设G_f经一次增广得到$G_{f'}$, 则$\forall u$, $\delta_f(s,u) \le \delta_{f'}(s,u)$

- 证明: 设v满足$\delta_f(s,v) > \delta_{f'}(s,v)$(1)且$\delta_{f'}(s,v)$最小(2),
  令u是$G_{f'}$中s到v最短路的父亲(s-->u->v是$G_{f'}$最短路径(3))
  则$(u,v) \notin G_f$(a), 从而G_f的增广路径经过(v,u)(4), 矛盾(b)

(a) $\delta_f(s,v) >_{(1)} \delta_{f'}(s,v) =_{(3)} \delta_{f'}(s,u) + 1 \ge_{(2)} \delta_f(s,u) + 1$

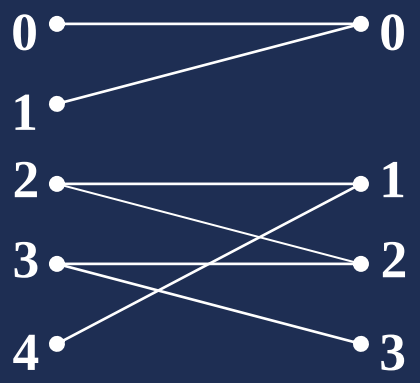
(b) $\delta_{f'}(s,v)-1 =_{(3)} \delta_{f'}(s,u) \ge_{(2)} \delta_f(s,u)$

$=_{(4)} \delta_f(s,v)+1 >_{(1)} \delta_{f'}(s,v)+1$

深度优先搜索匹配算法分析

```
…; memset(match, -1, sizeof(match);
    for (i = 0; i < boy; i++) {
    memset(color, 0, sizeof(color));
    if (dfs(i)) ans++;     } …
```

```
int dfs(int left) {
    int j;
    for (j = 0; j < girl; j++) {
    if (!color[j] && map[left][j]) {
        color[j] = 1;
    if (match[j] == -1 || dfs(match[j]))
        {  match[j] = left;
        return 1;
        }  }  }
```



match	0	1	2	3
初始	-1	-1	-1	-1
dfs(0)	0	-1	-1	-1
dfs(1)	0	-1	-1	-1
dfs(2)	0	2	-1	-1
dfs(3)	0	2	3	-1
dfs(4)	0	4	2	3

匹配算法综述

- 定理:对二分图, ρ(G)=c(G).

  König[1931], Egerváry[1931]

- 二分图匹配:匈牙利算法, $O(|V||E|) \rightarrow |V|^{1/2}|E|$
- 加权二分图匹配: 匈牙利算法(或KM算法),

  Kuhn[1955], Munkres[1957]

- 一般图匹配: 开花算法, Edmonds' Blossom Alg.

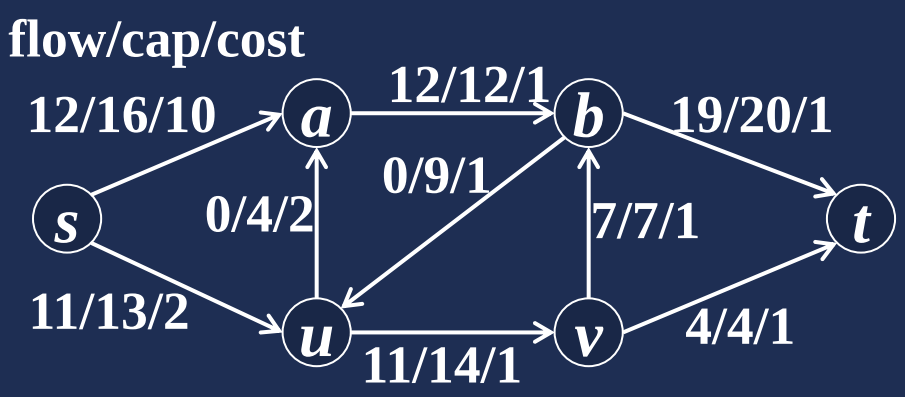
  1965, $O(|V|^4) \rightarrow O(|V|^{1/2}|E|)$

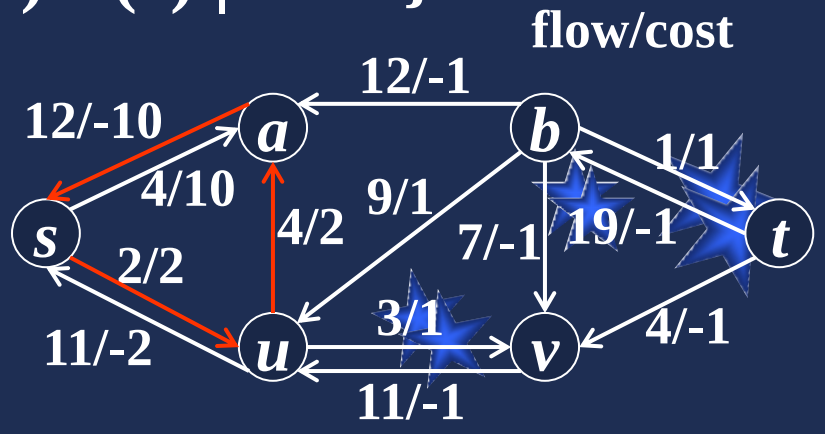
- 一般图覆盖和独立集, NP完全问题,

  目前无多项式时间算法

最小费用最大流

- 有向图G=(V,E), 容量cap: V×V→R非负, 源s, 汇t,
  - 每条边的单位流量花费cost: V×V→R非负,
  - 求容量限制下, 从s到t最大流量下最小花费
    - 若(u,v)∈E, 则cost(v,u) = - cost(u,v)
- 最小费用最大流问题的最优性定理: G的最大流f是G的最小费用流 ⇔ G_f 没有负费用圈
  - cost(f) = sum { cost(e)×f(e) | e∈E }

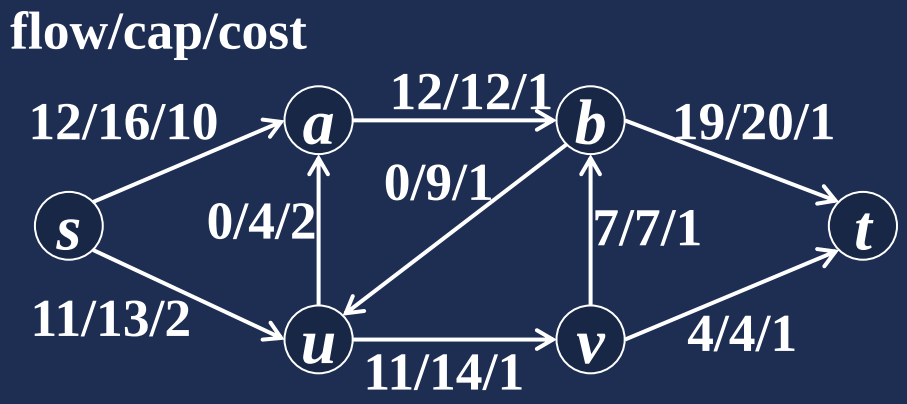


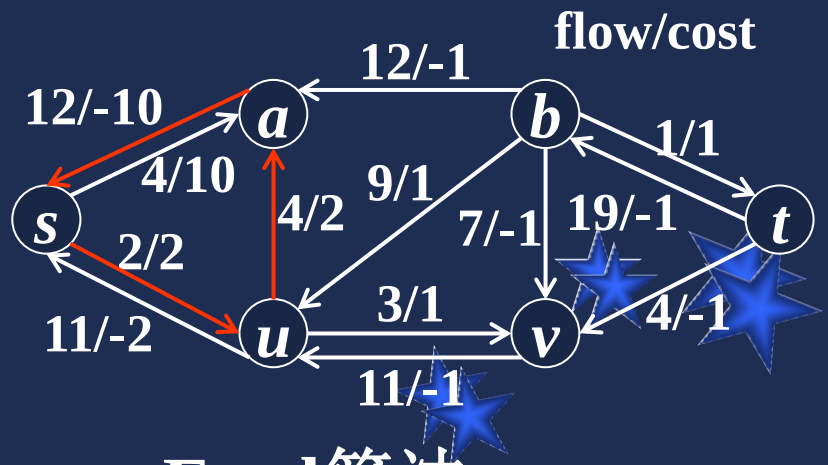


最小费用最大流算法

- 有向图G=(V,E), 容量cap: V×V→R非负, 源s, 汇t,
  - 每条边的花费cost: V×V→R非负,
- 求容量限制下, 从s到t最大流量下最小花费

1. 用最大流算法构造最大流f
2. 当G_f中存在负费用圈,
3. 沿负费用圈增流得新流f





寻找负费用圈: 最短路Bellman-Ford算法