计算机问题求解 – 论题3-13

- 最大流算法

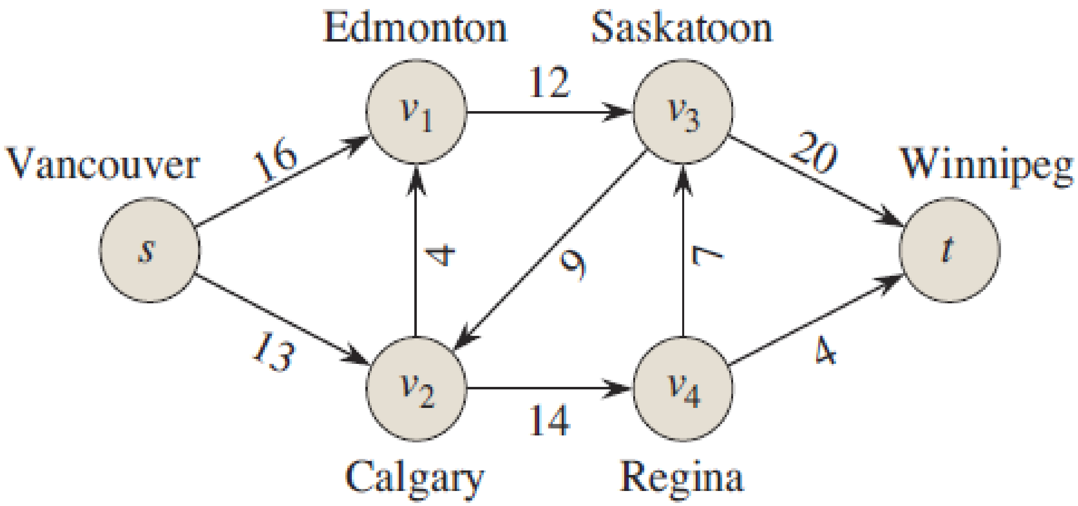
2016年11月30日

Lucky Puck Company’s Trucking Problem

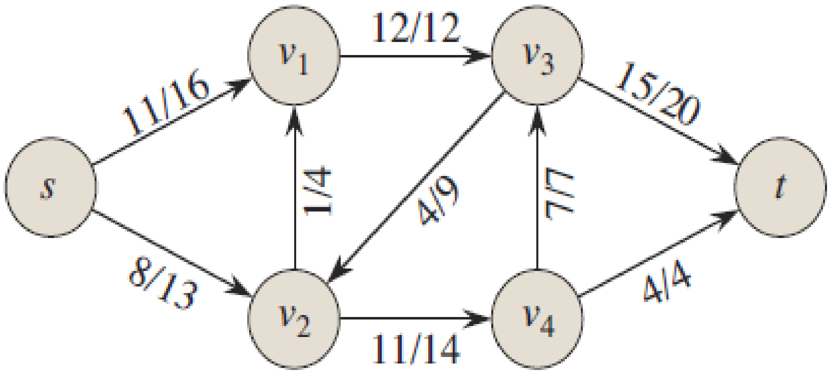
Lucky公司生产冰球，其工厂在温哥华，仓库在温尼泊。公司委托物流公司运输。物流公司经营固定线路网，可能经过多个中间城市。分配给Lucky公司的任意两个城市间的最大运输量是固定的。如果Lucky公司是按运输量确定生产量，它如何计划它每天最大产量？

问题1:

如何建立解决这个问题的模型?



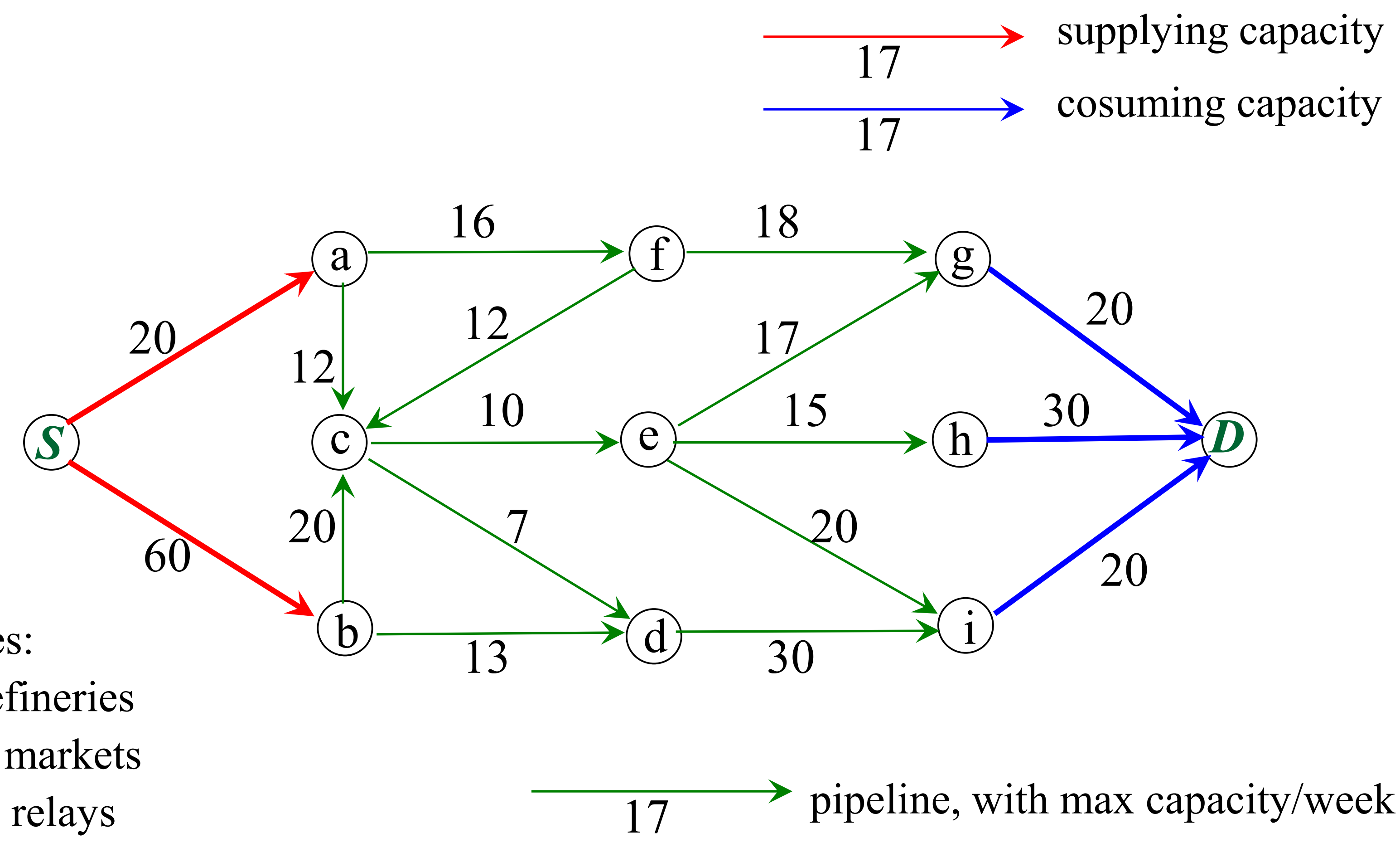
问题2:运输方案有多种,最优方案是什么?图中的方案是最优的吗?



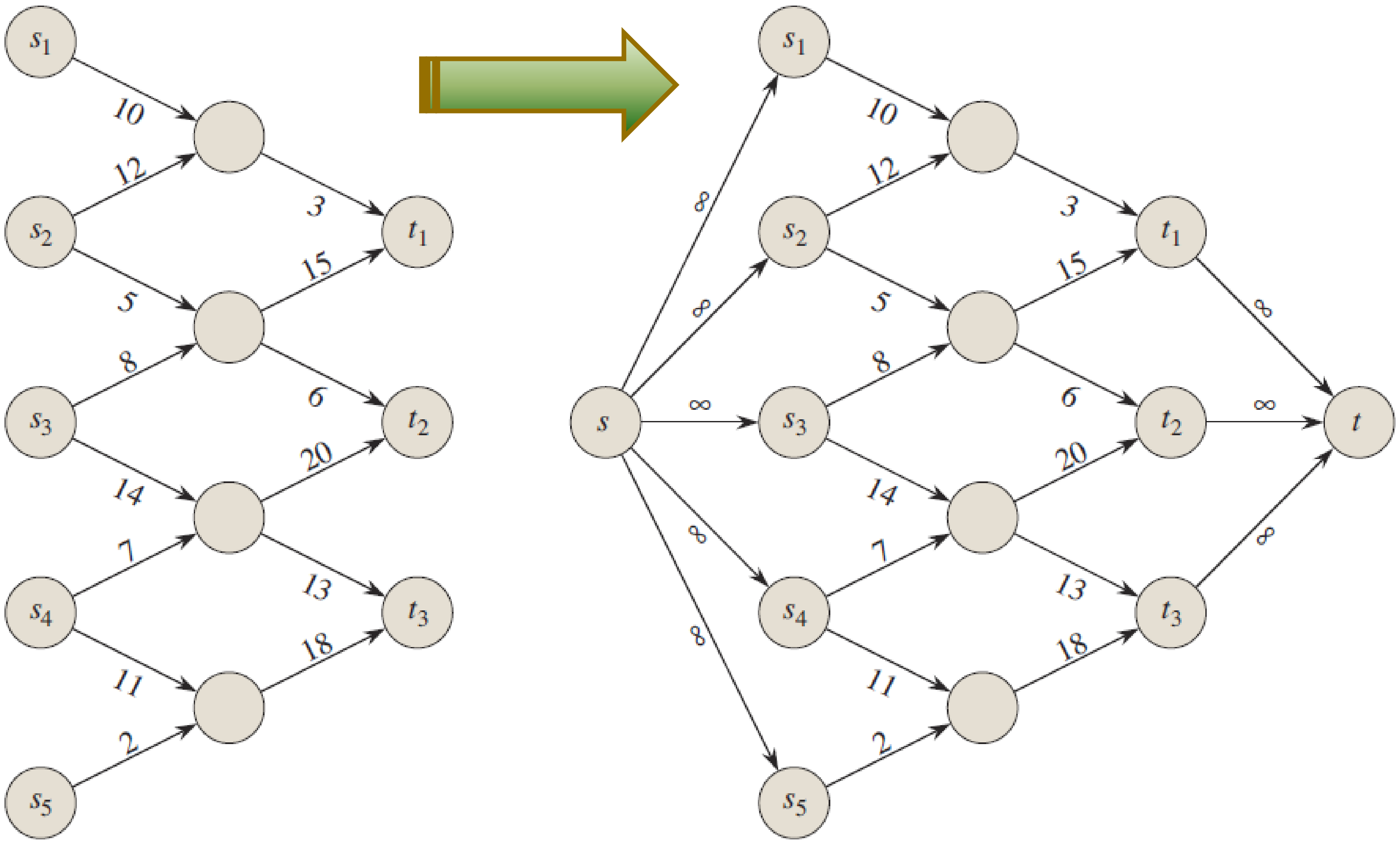
问题3：

如果工厂与仓库都不止一个，该怎么办？

A Model of Oil Supply



Vertices:
a, *b*: refineries
g, *h*, *i*: markets
others: relays


s_1
s_2
s_3
s_4
s_5
t_1
t_2
t_3
10
12
3
15
5
8
6
20
14
7
13
18
11
2
s
t
∞

严格的数学模型

We are now ready to define flows more formally. Let $G = (V, E)$ be a flow network with a capacity function c. Let s be the source of the network, and let t be the sink. A ***flow*** in G is a real-valued function $f : V \times V \rightarrow \mathbb{R}$ that satisfies the following two properties:

Capacity constraint:

Flow conservation: For all $u \in V - \{s, t\}$, we require

When $(u, v) \notin E$, there can be no flow from u to v, and $f(u, v) = 0$.

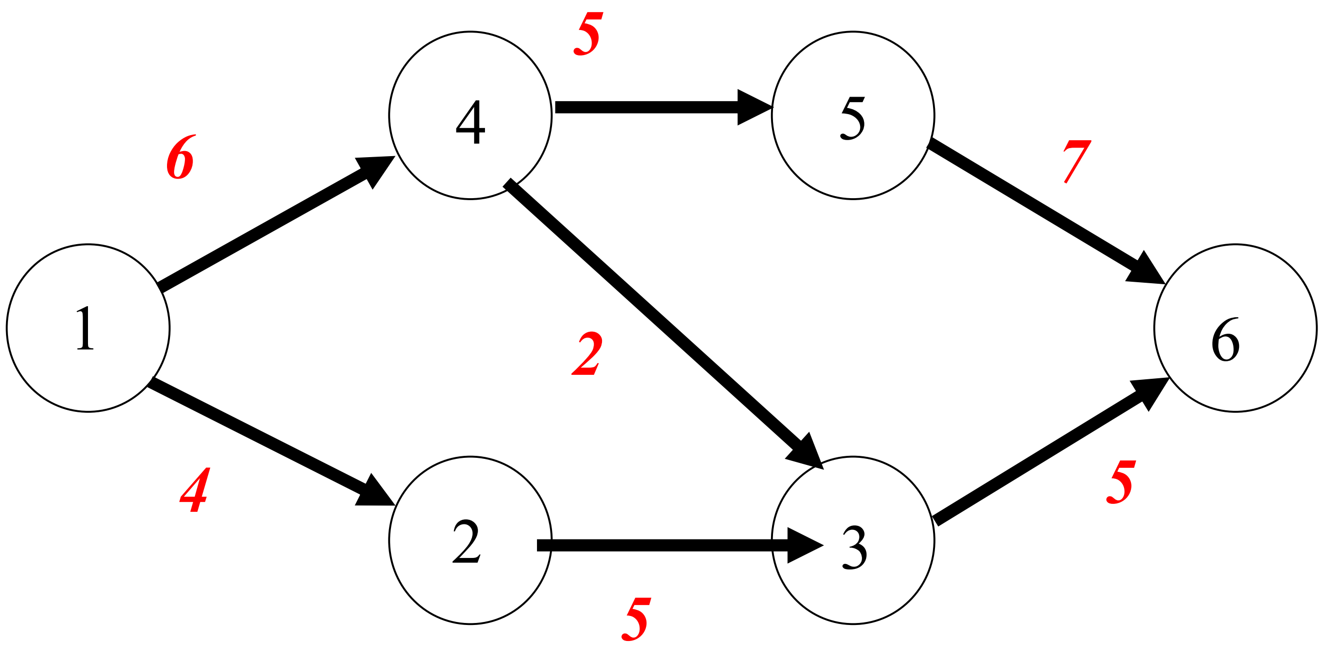
问题4:

什么叫一个flow的“value”?

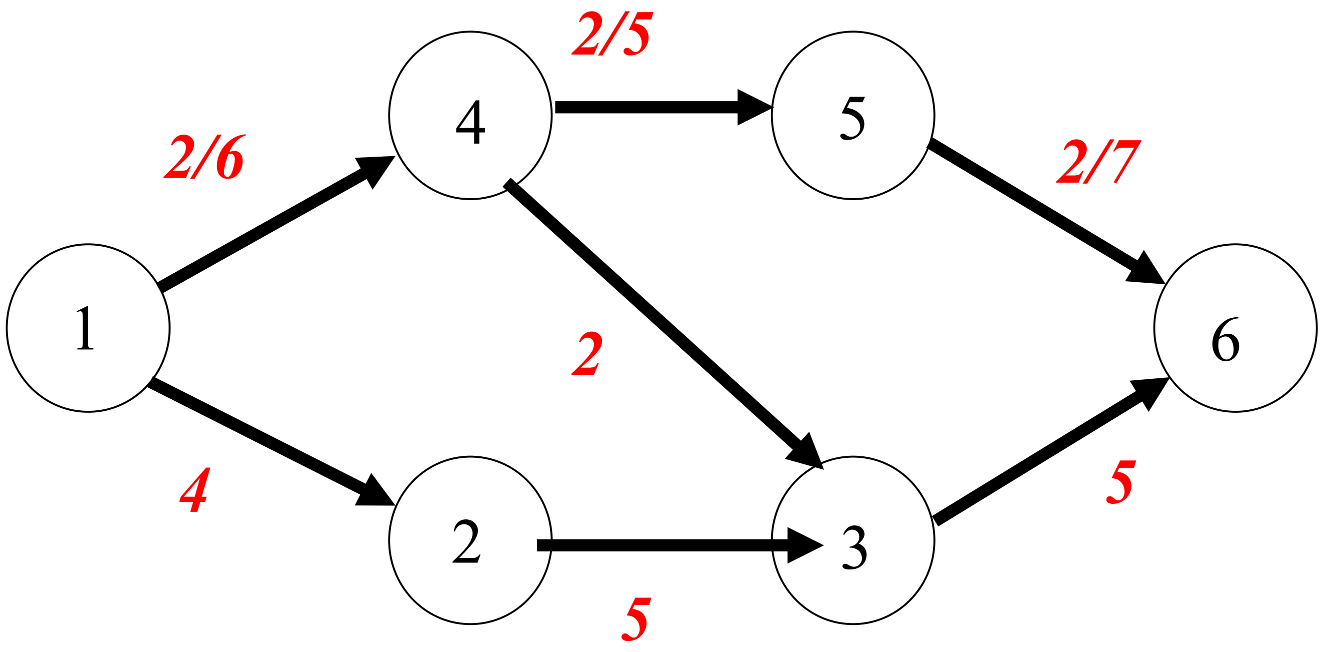
$$|f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s)$$

什么是最大流问题?

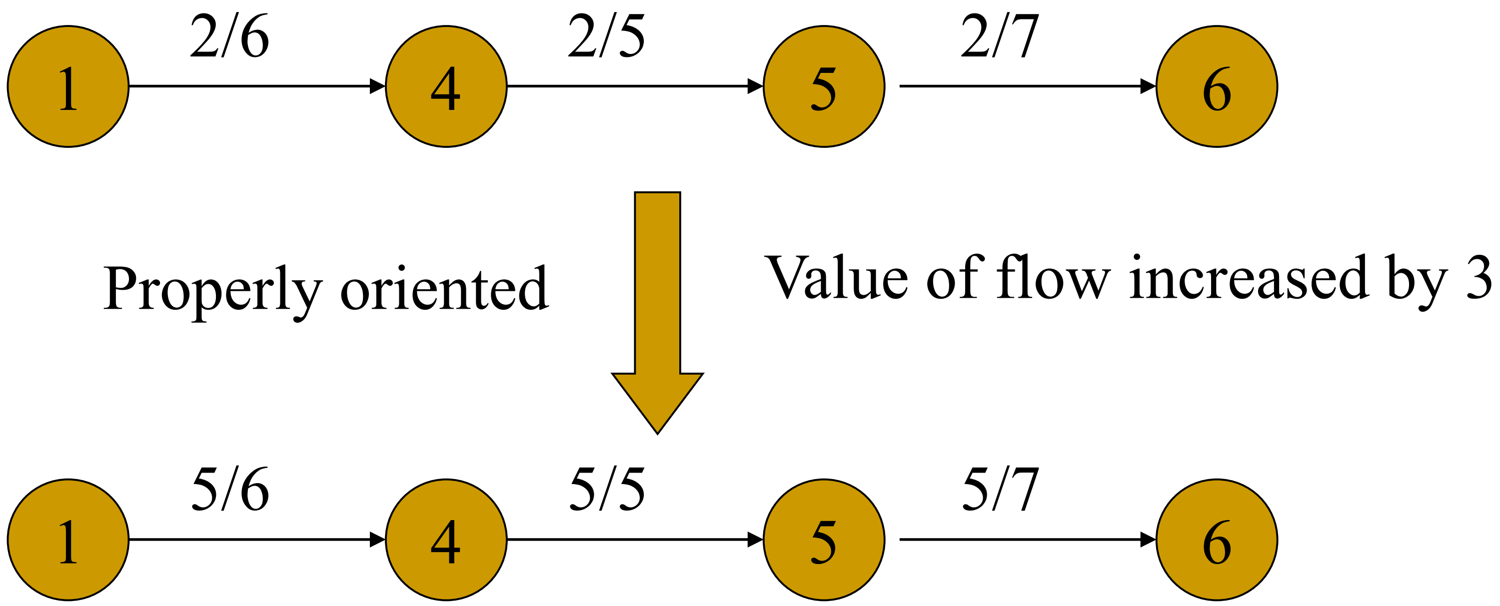
How to get the maximum flow?



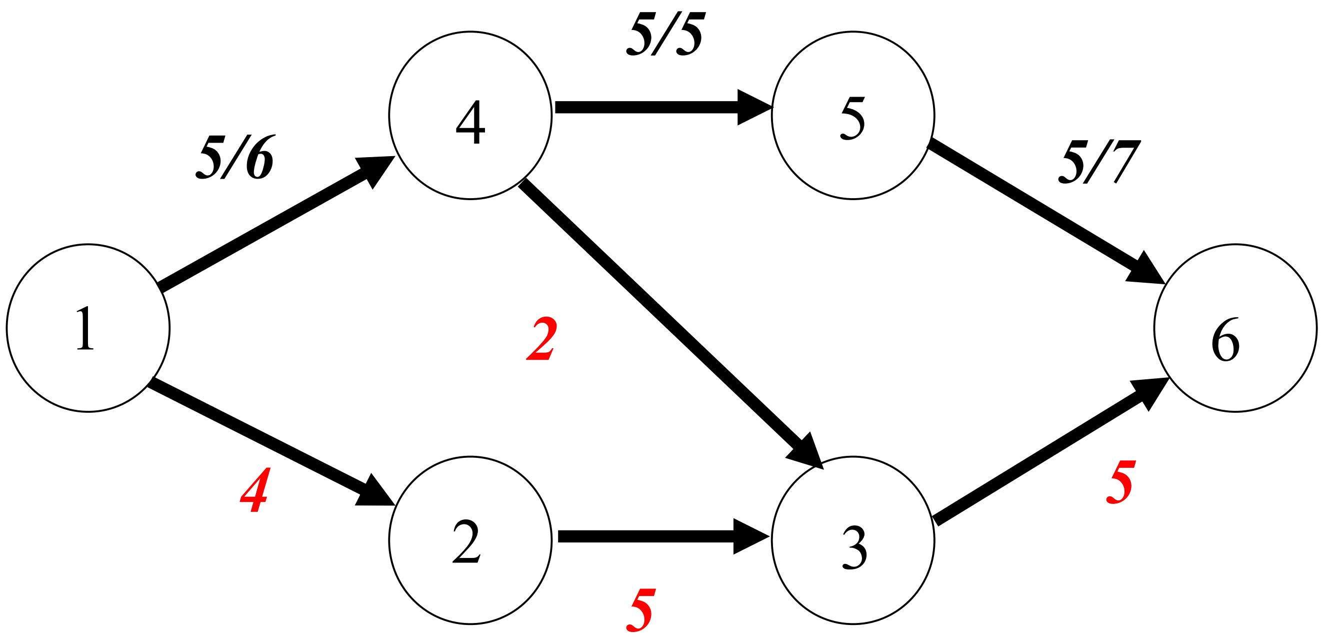
How to get the maximum flow?



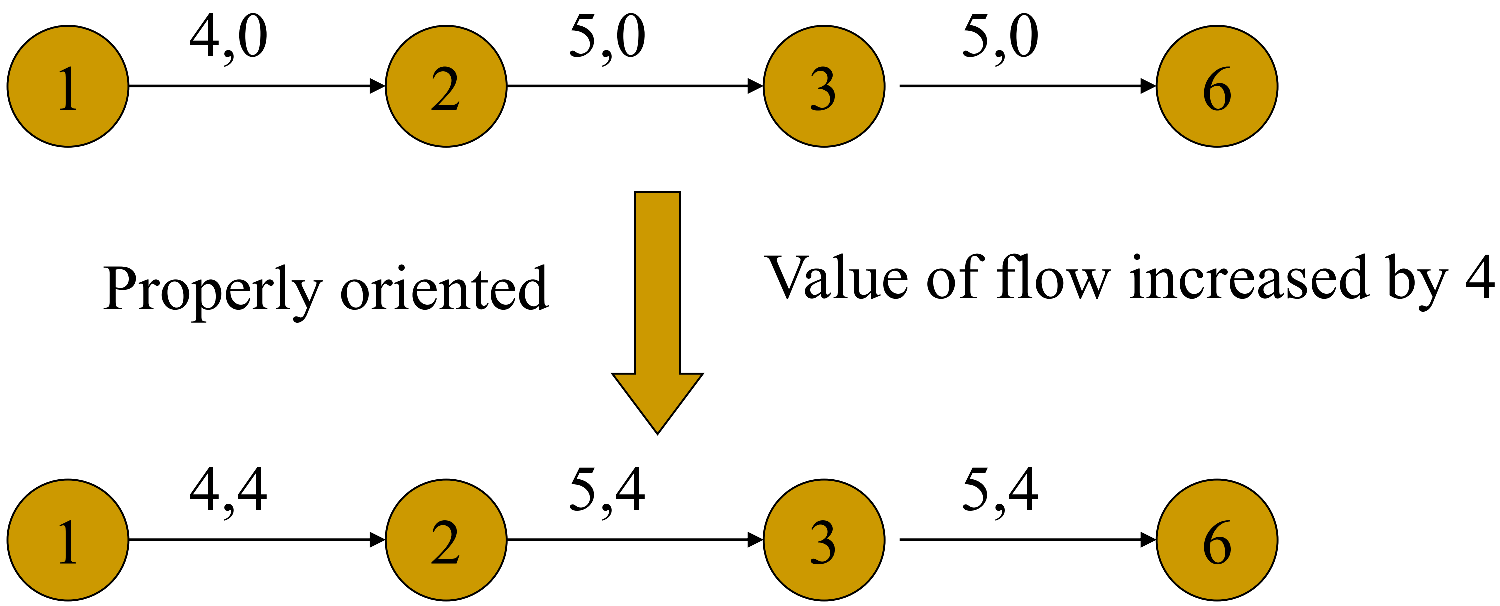
Path1 in N


1
2/6
4
2/5
5
2/7
6
Properly oriented
Value of flow increased by 3
1
5/6
4
5/5
5
5/7
6

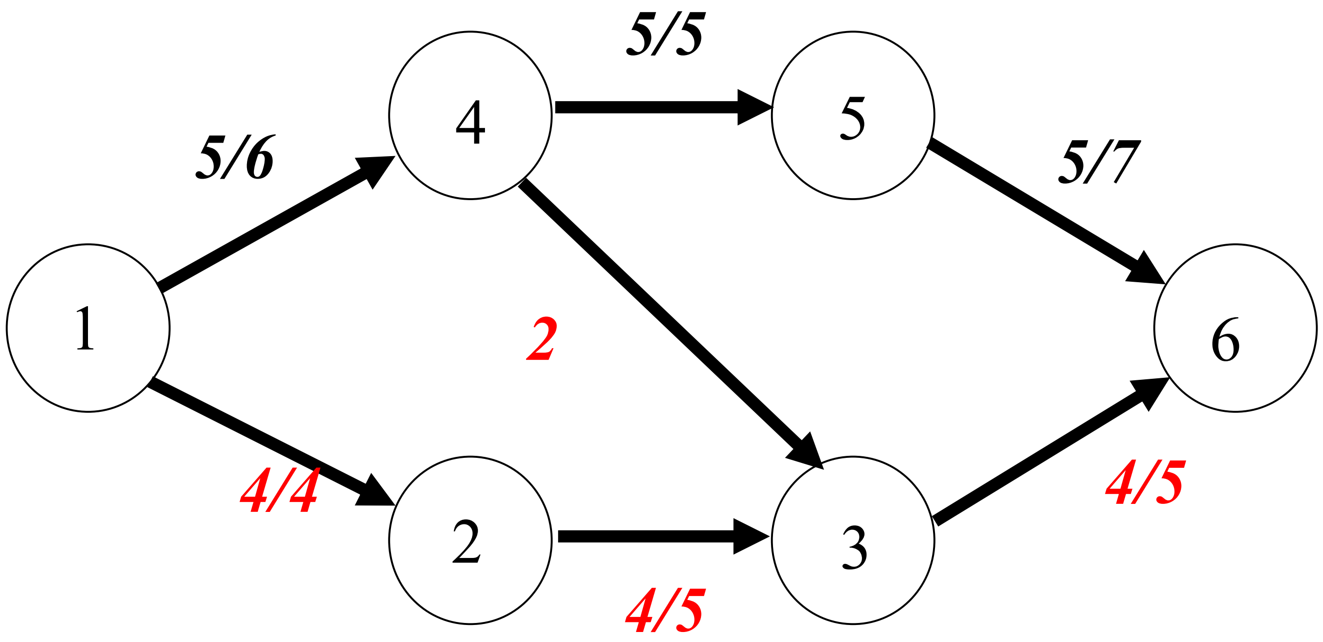
How to get the maximum flow?



Path2 in N

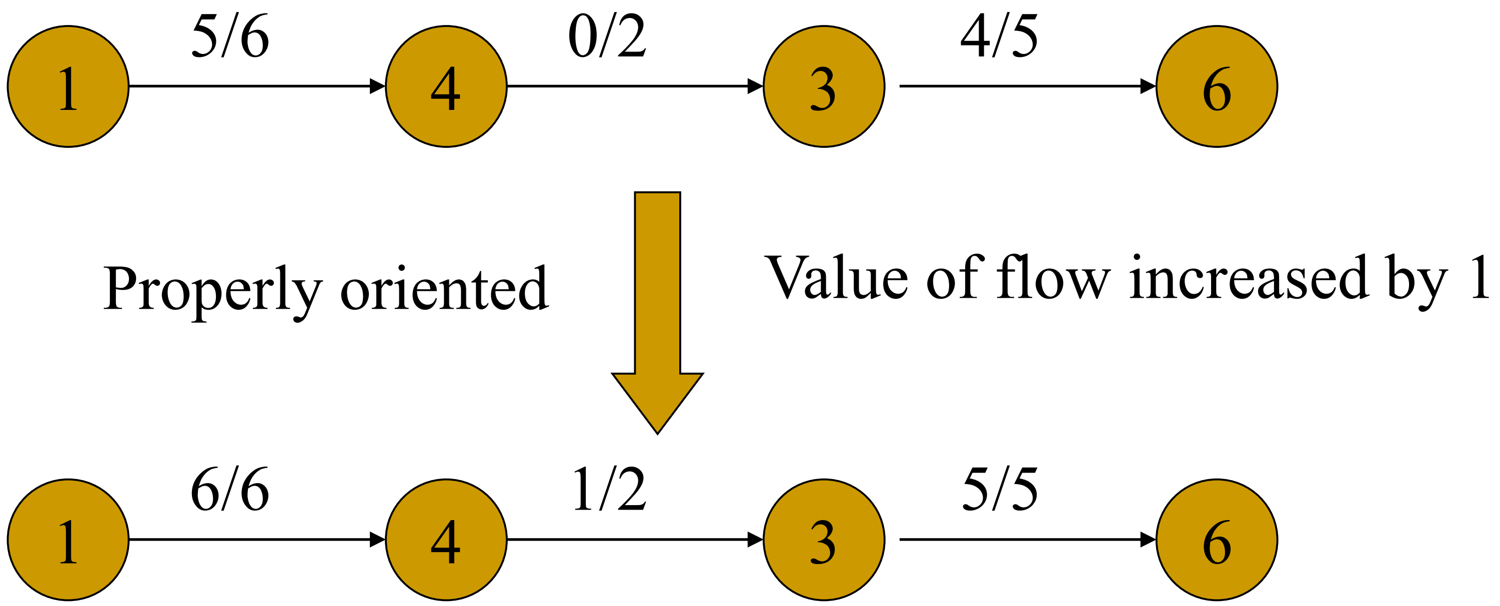

4,0
5,0
5,0
1
2
3
6
Properly oriented
Value of flow increased by 4
4,4
5,4
5,4
1
2
3
6

How to get the maximum flow?

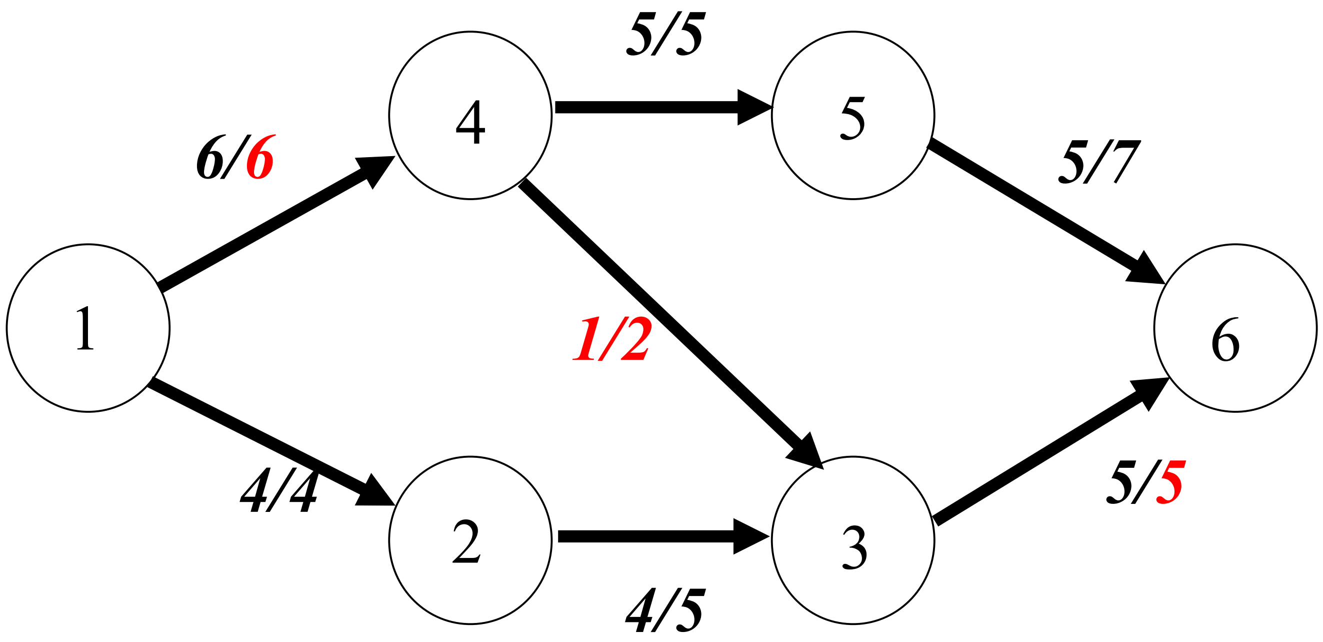


No other path?

Path3 in N


1
5/6
4
0/2
3
4/5
6
Properly oriented
Value of flow increased by 1
1
6/6
4
1/2
3
5/5
6

No other path!



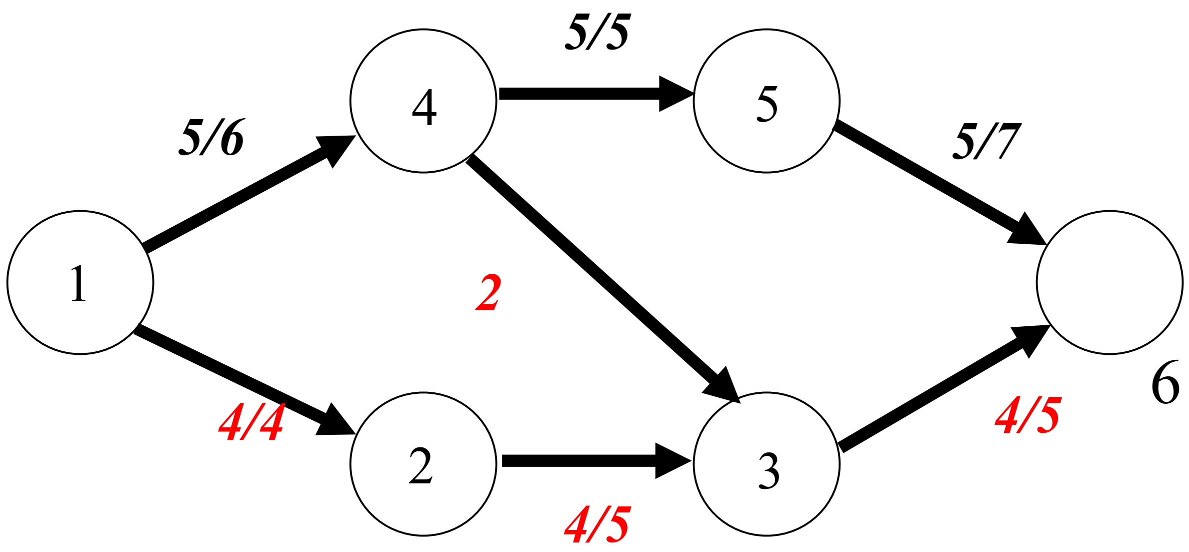
Find all the path and increase its flow value to the max!

FORD-FULKERSON-METHOD(G, s, t)

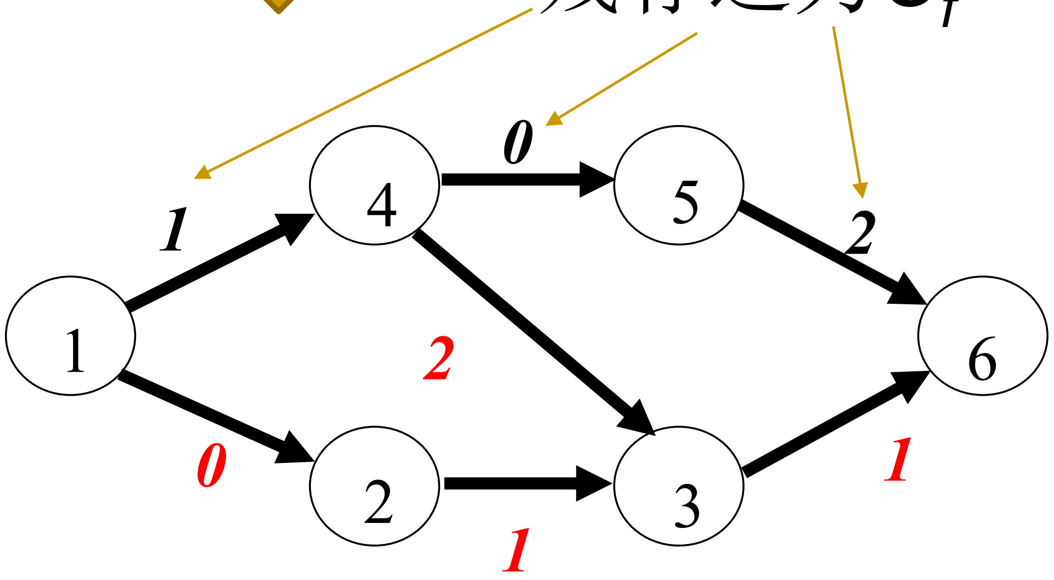
initialize flow f to 0
while there exists an augmenting path p in the residual network G_f
    augment flow f along p
return f

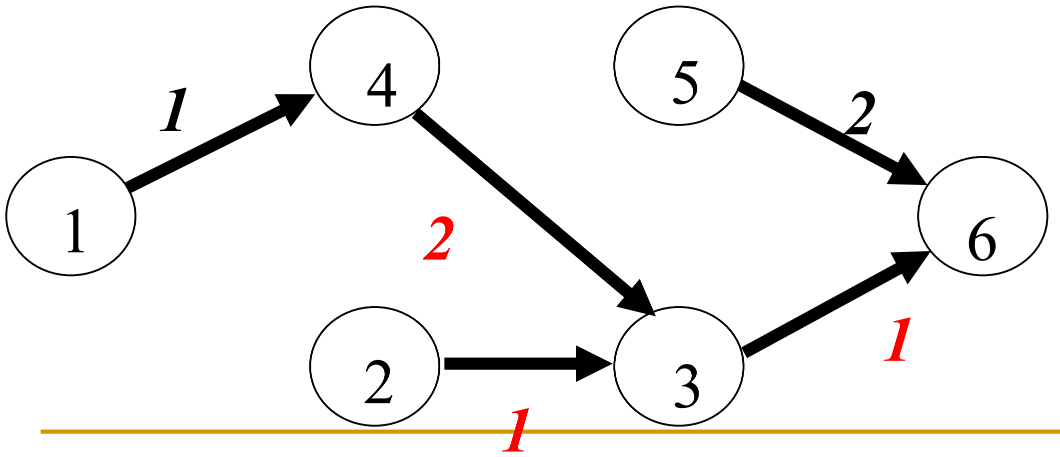
找到所有的augmenting path是方法的关键所在!

再看这个流网络中的流f:



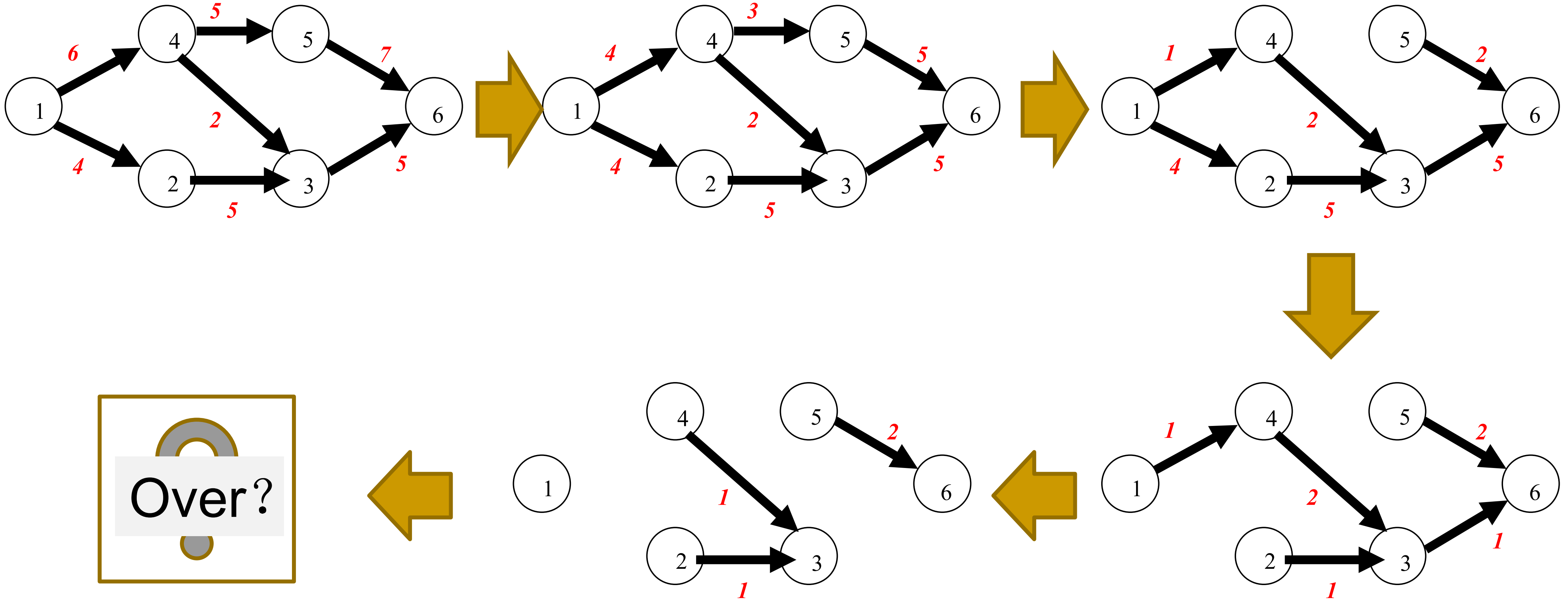
当前流f下的残存运力C_f





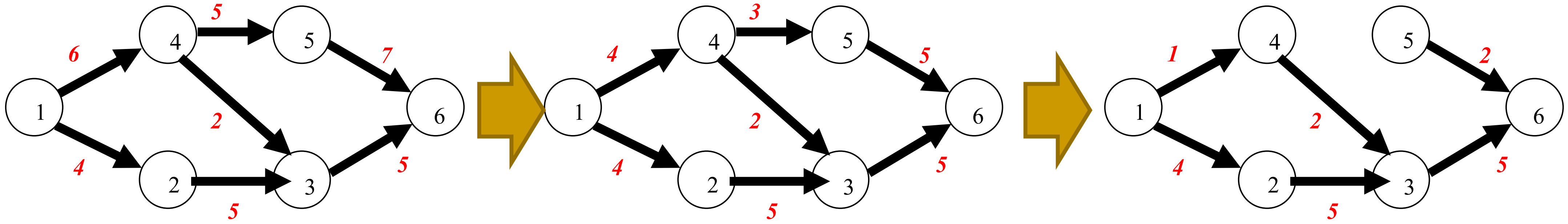
当前流f下的残存网络G_f

借助残存运力/网络概念，再看上述寻找过程



A flow in a residual network provides a roadmap for adding flow to the original flow network. If f is a flow in G and f' is a flow in the corresponding residual network G_f, we define $f \uparrow f'$, the ***augmentation*** of flow f by f', to be a function from $V \times V$ to $\mathbb{R}$, defined by

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v) + f'(u,v) & \text{if } (u,v) \in E , \\ 0 & \text{otherwise} . \end{cases}$$



Lemma 26.1

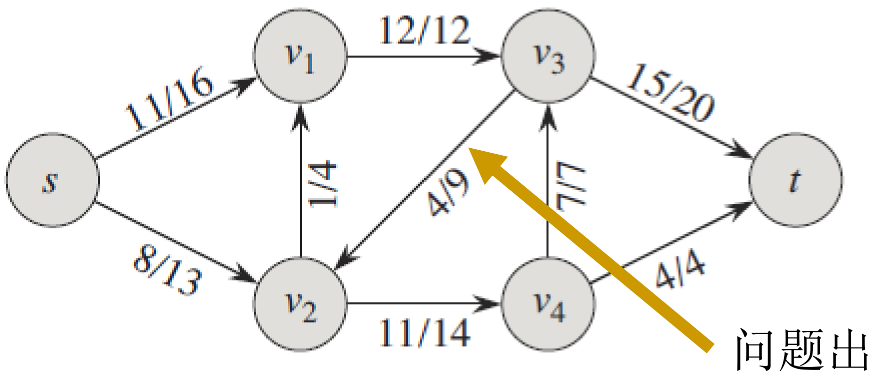
Let $G = (V, E)$ be a flow network with source s and sink t, and let f be a flow in G. Let G_f be the residual network of G induced by f, and let f' be a flow in G_f. Then the function $f \uparrow f'$ defined in equation (26.4) is a flow in G with value $|f \uparrow f'| = |f| + |f'|$.

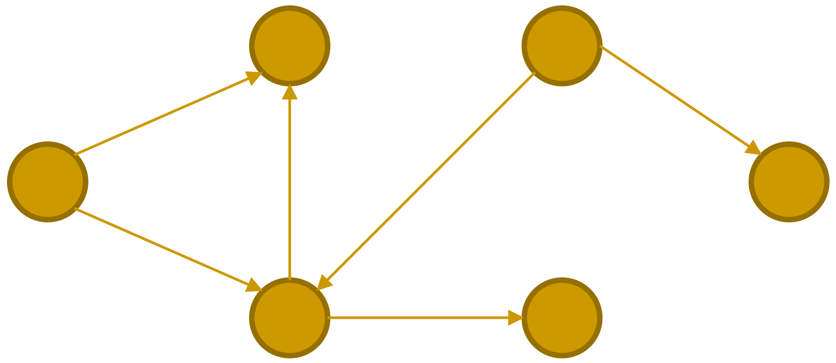
证明要点：

1，证明函数是一个流：符合流的两个特性

2，证明 $|f \uparrow f'| = |f| + |f'|$.

我们找到的方法一定正确吗？





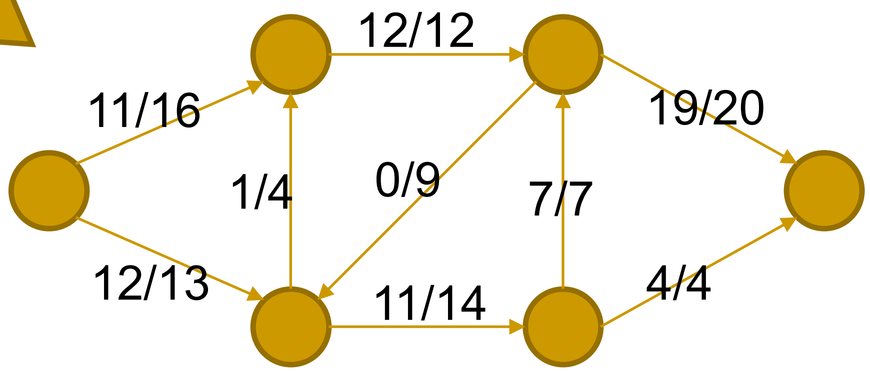
问题出在什么地方？

这是最大的流了吗？

这种决策在增广流过程中能保证不出现吗？

我们有办法还是在“残存”模型中“纠正”这个错误吗？

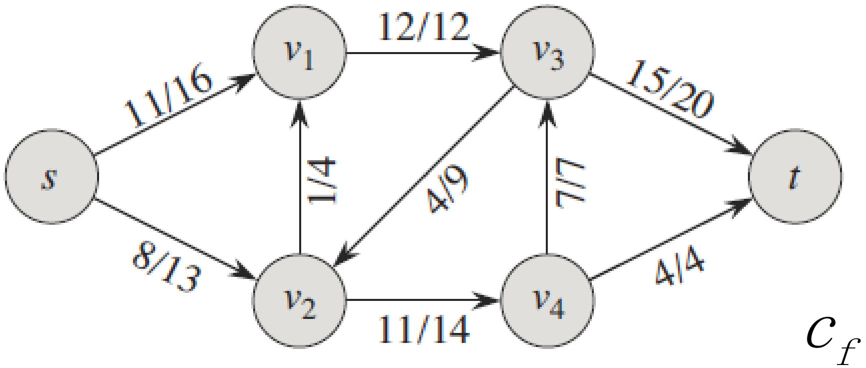
No！



residual capacity

define the ***residual capacity*** $c_f(u,v)$ by

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E\,, \\ f(v,u) & \text{if } (v,u) \in E\,, \\ 0 & \text{otherwise}\,. \end{cases}$$



$c_f(v_3, v_2) = ?$

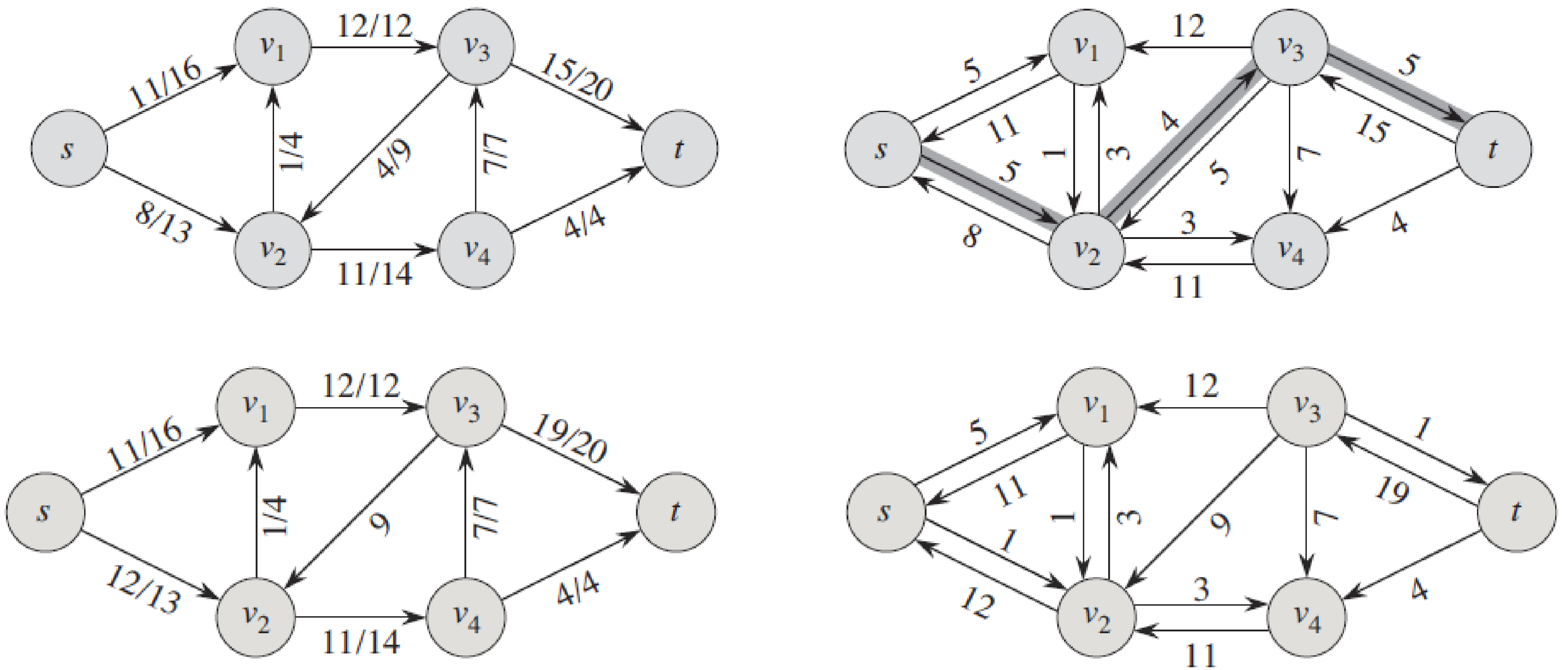
$c_f(v_2, v_3) = ?$

问题5：我们为什么要如此定义residual capacity?

Residual Network

Given a flow network $G = (V, E)$ and a flow f, the ***residual network*** of G induced by f is $G_f = (V, E_f)$, where

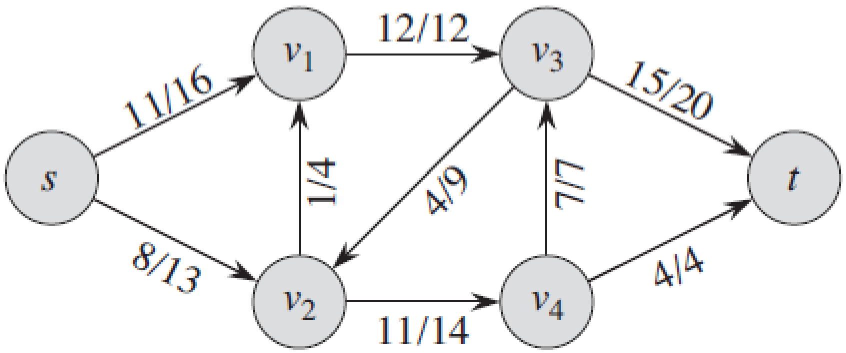
$$E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\} . \tag{26.3}$$

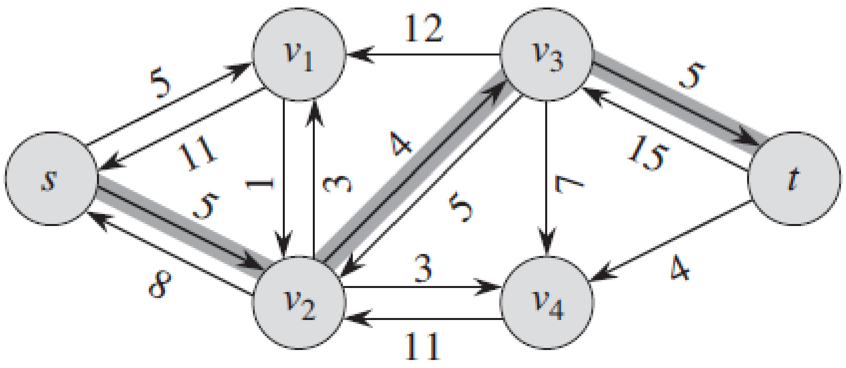


残存网络中找不到源到目的地的路径，所有增广路确定全部找到？

A flow in a residual network provides a roadmap for adding flow to the original flow network. If f is a flow in G and f' is a flow in the corresponding residual network G_f, we define $f \uparrow f'$, the ***augmentation*** of flow f by f', to be a function from $V \times V$ to $\mathbb{R}$, defined by

$$(f \uparrow f')(u,v) = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) & \text{if } (u,v) \in E \,, \\ 0 & \text{otherwise} \,. \end{cases} \tag{26.4}$$





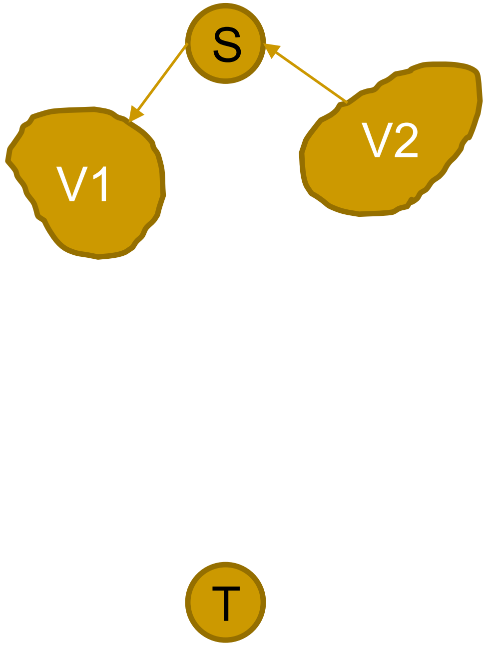
Lemma 26.1

Let $G = (V, E)$ be a flow network with source s and sink t, and let f be a flow in G. Let G_f be the residual network of G induced by f, and let f' be a flow in G_f. Then the function $f \uparrow f'$ defined in equation (26.4) is a flow in G with value $|f \uparrow f'| = |f| + |f'|$.

证明要点：

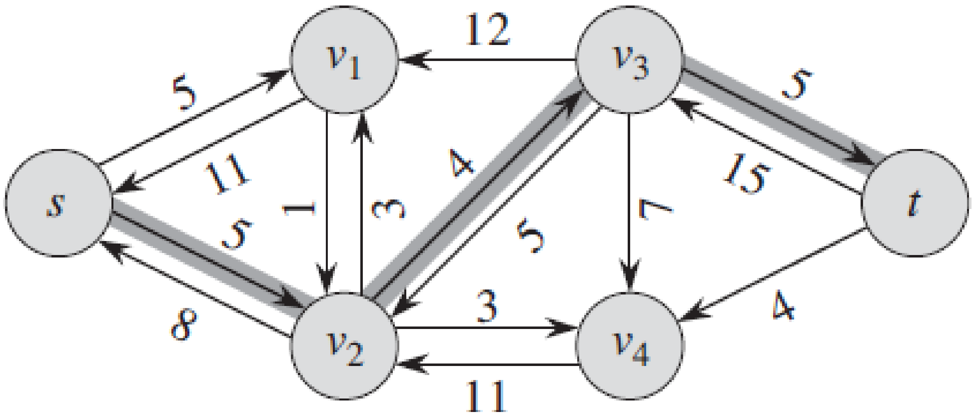
1，证明函数是一个流：符合流的两个特性

2，证明 $|f \uparrow f'| = |f| + |f'|$.



$$
\begin{aligned}
|f \uparrow f'| &= \sum_{v \in V} (f \uparrow f')(s, v) - \sum_{v \in V} (f \uparrow f')(v, s) \\
&= \sum_{v \in V_1} (f \uparrow f')(s, v) - \sum_{v \in V_2} (f \uparrow f')(v, s) , \\
&= \sum_{v \in V_1} (f(s, v) + f'(s, v) - f'(v, s)) - \sum_{v \in V_2} (f(v, s) + f'(v, s) - f'(s, v)) \\
&= \sum_{v \in V_1} f(s, v) + \sum_{v \in V_1} f'(s, v) - \sum_{v \in V_1} f'(v, s) \\
&\qquad - \sum_{v \in V_2} f(v, s) - \sum_{v \in V_2} f'(v, s) + \sum_{v \in V_2} f'(s, v) \\
&= \sum_{v \in V_1} f(s, v) - \sum_{v \in V_2} f(v, s) \\
&\qquad + \sum_{v \in V_1} f'(s, v) + \sum_{v \in V_2} f'(s, v) - \sum_{v \in V_1} f'(v, s) - \sum_{v \in V_2} f'(v, s) \\
&= \sum_{v \in V_1} f(s, v) - \sum_{v \in V_2} f(v, s) + \sum_{v \in V_1 \cup V_2} f'(s, v) - \sum_{v \in V_1 \cup V_2} f'(v, s) . \qquad (26.6) \\
&= \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) + \sum_{v \in V} f'(s, v) - \sum_{v \in V} f'(v, s) \\
&= |f| + |f'| .
\end{aligned}
$$

增广路径



$$c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$$

问题7：如果在residual network中发现了一条s,t路径，是否一定可以将流网络中的流进行扩大?

问题8：If not，是否意味着最大流已经出现?

答案是肯定的：

Corollary 26.3

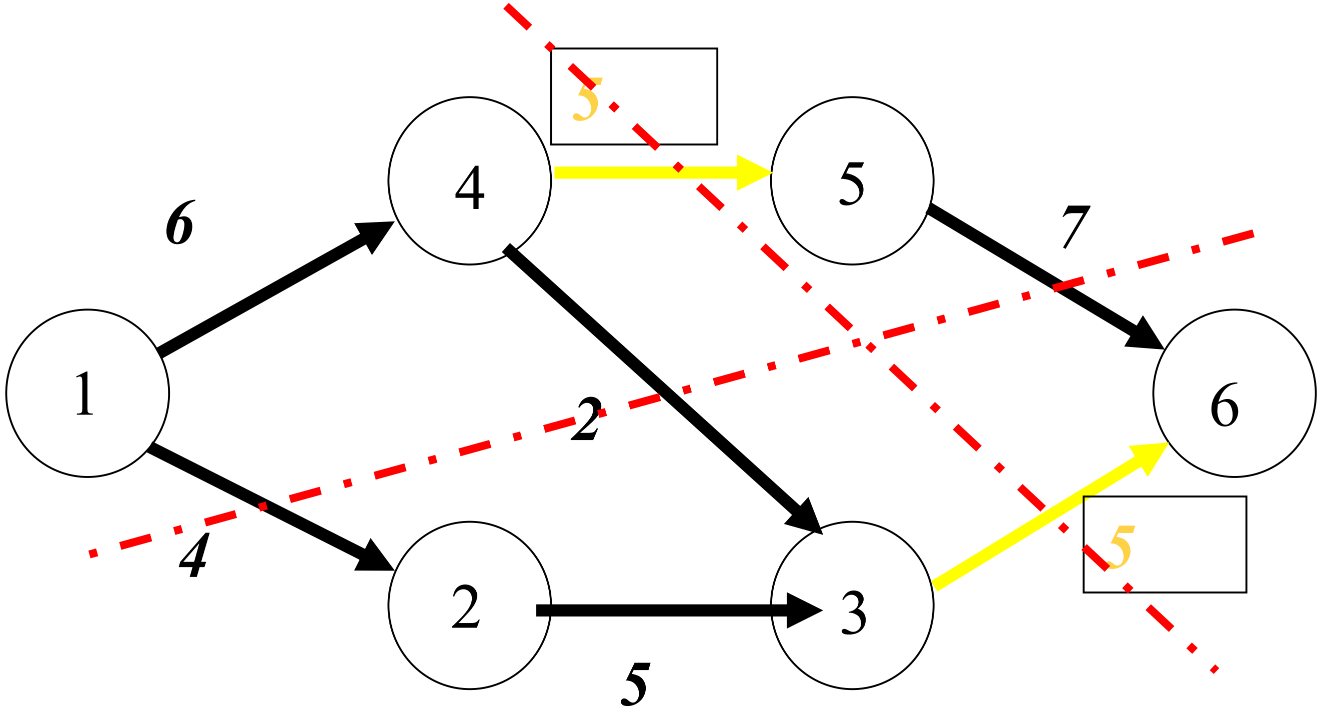
Let $G = (V, E)$ be a flow network, let f be a flow in G, and let p be an augmenting path in G_f. Let f_p be defined as in equation (26.8), and suppose that we augment f by f_p. Then the function $f \uparrow f_p$ is a flow in G with value $|f \uparrow f_p| = |f| + |f_p| > |f|$.

Theorem 26.6 (Max-flow min-cut theorem)

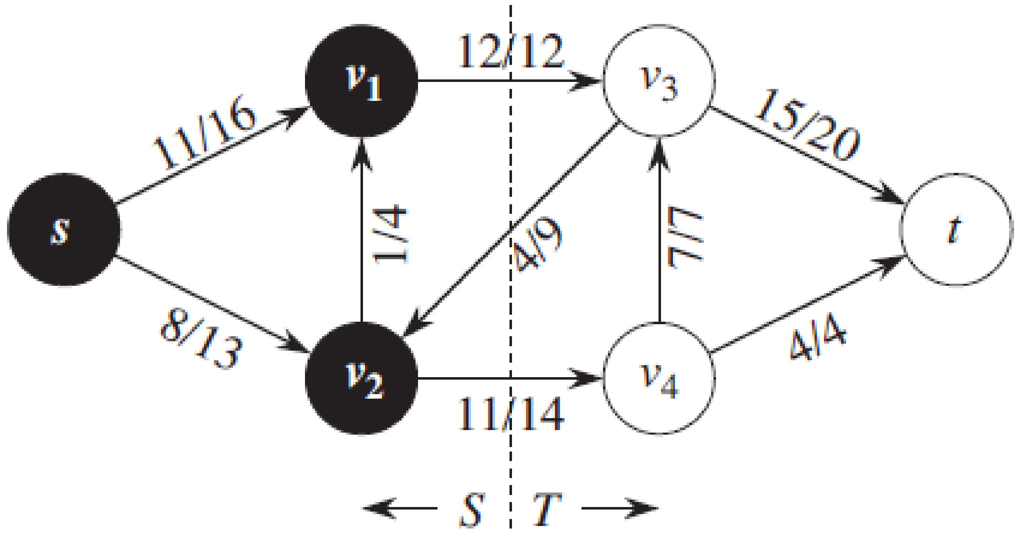
If f is a flow in a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:

1. f is a maximum flow in G.
2. The residual network G_f contains no augmenting paths.
3. $|f| = c(S, T)$ for some cut (S, T) of G.

问题10：任意的流值，都不会超过任意的割容量?



流网络的割



$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) .$$

问题9：任何一个割的净流量是否一定等于*f*的值？

Yes:

Lemma 26.4

Let f be a flow in a flow network G with source s and sink t, and let (S, T) be any cut of G. Then the net flow across (S, T) is $f(S, T) = |f|$.

流量守恒

证明：

$$|f| = \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) + \boxed{\sum_{u\in S-\{s\}} \left(\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u) \right)} .$$

$$\begin{aligned} |f| &= \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) + \sum_{u\in S-\{s\}} \sum_{v\in V} f(u,v) - \sum_{u\in S-\{s\}} \sum_{v\in V} f(v,u) \\ &= \sum_{v\in V} \left(f(s,v) + \sum_{u\in S-\{s\}} f(u,v) \right) - \sum_{v\in V} \left(f(v,s) + \sum_{u\in S-\{s\}} f(v,u) \right) \\ &= \sum_{v\in V} \sum_{u\in S} f(u,v) - \sum_{v\in V} \sum_{u\in S} f(v,u) . \end{aligned}$$

$$
\begin{aligned}
|f| &= \sum_{v\in S}\sum_{u\in S} f(u,v) + \sum_{v\in T}\sum_{u\in S} f(u,v) - \sum_{v\in S}\sum_{u\in S} f(v,u) - \sum_{v\in T}\sum_{u\in S} f(v,u) \\
&= \sum_{v\in T}\sum_{u\in S} f(u,v) - \sum_{v\in T}\sum_{u\in S} f(v,u) \\
&\qquad + \left(\sum_{v\in S}\sum_{u\in S} f(u,v) - \sum_{v\in S}\sum_{u\in S} f(v,u)\right).
\end{aligned}
$$

S和T是V的划分

$f(x,y)$重复出现，结果为0

The two summations within the parentheses are actually the same, since for all vertices $x, y \in V$, the term $f(x,y)$ appears once in each summation. Hence, these summations cancel, and we have

$$
\begin{aligned}
|f| &= \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u) \\
&= f(S,T).
\end{aligned}
$$

■

Corollary 26.5

The value of any flow f in a flow network G is bounded from above by the capacity of any cut of G.

Proof Let (S, T) be any cut of G and let f be any flow. By Lemma 26.4 and the capacity constraint,

$$\begin{aligned}
|f| &= f(S,T) \\
&= \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) \\
&\leq \sum_{u \in S} \sum_{v \in T} f(u,v) \\
&\leq \sum_{u \in S} \sum_{v \in T} c(u,v) \\
&= c(S,T)\ .
\end{aligned}$$

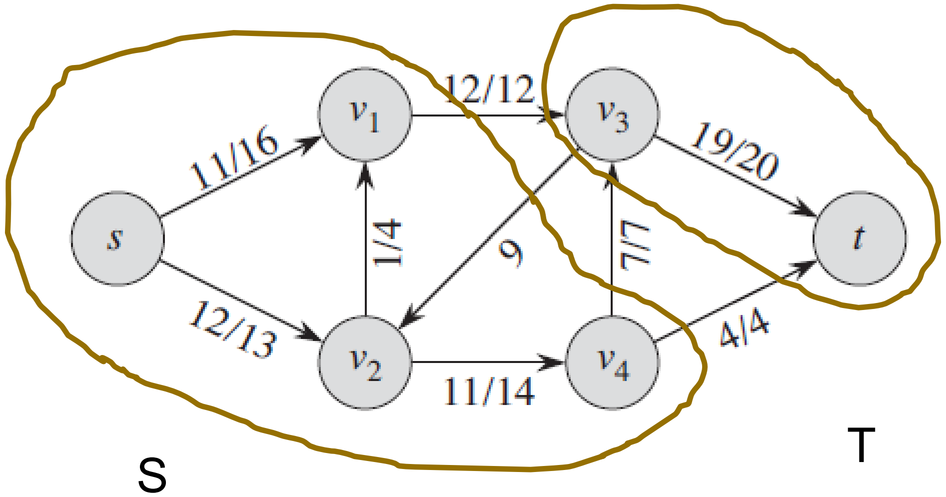
■

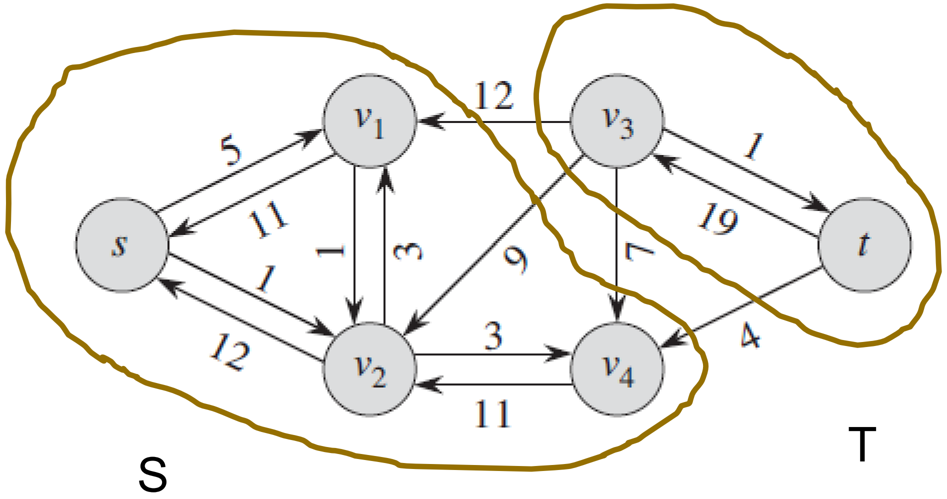
Theorem 26.6 (Max-flow min-cut theorem)

If f is a flow in a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:

1. f is a maximum flow in G.
2. The residual network G_f contains no augmenting paths.
3. $|f| = c(S, T)$ for some cut (S, T) of G.

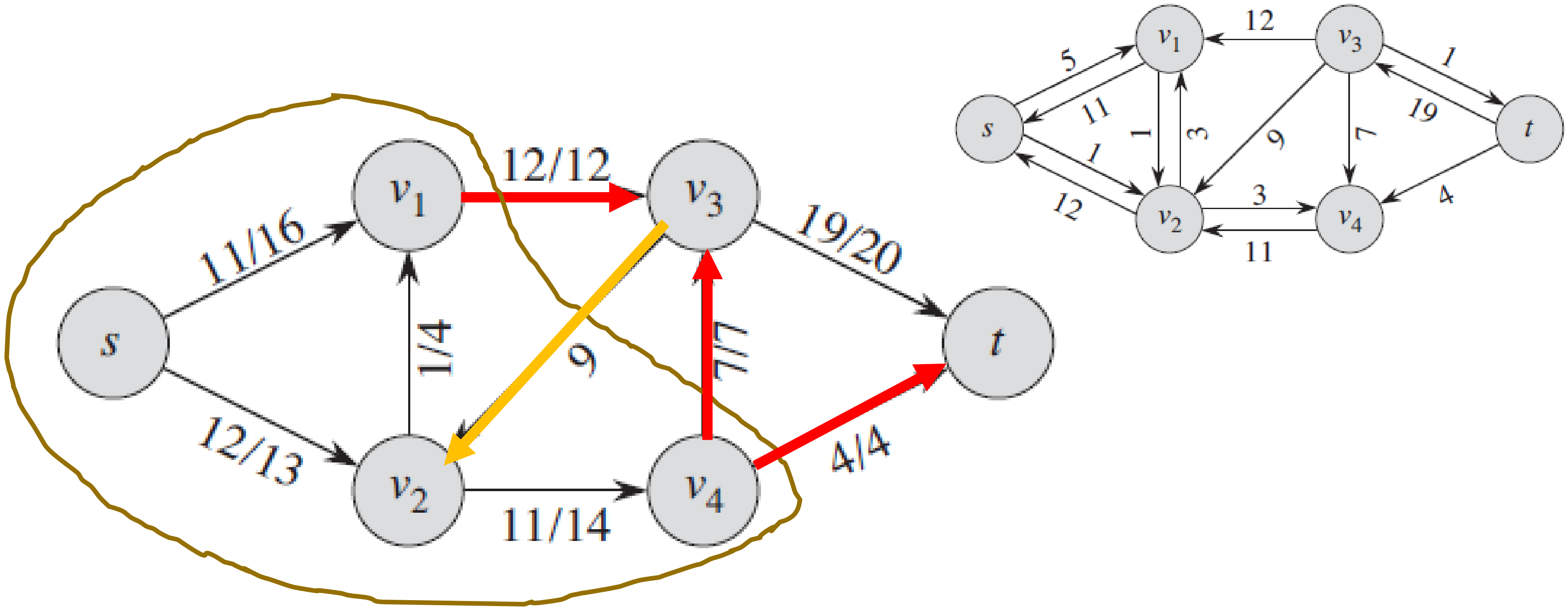
证明要点：找不到增广路径时，残存网络必定将网络进行了切割，这个切割的容量恰好就是这个流的值





$$
\begin{aligned}
f(S,T) &= \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{v\in T}\sum_{u\in S} f(v,u) \\
&= \sum_{u\in S}\sum_{v\in T} c(u,v) - \sum_{v\in T}\sum_{u\in S} 0 \\
&= c(S,T) .
\end{aligned}
$$

By Lemma 26.4, therefore, $|f| = f(S,T) = c(S,T)$.



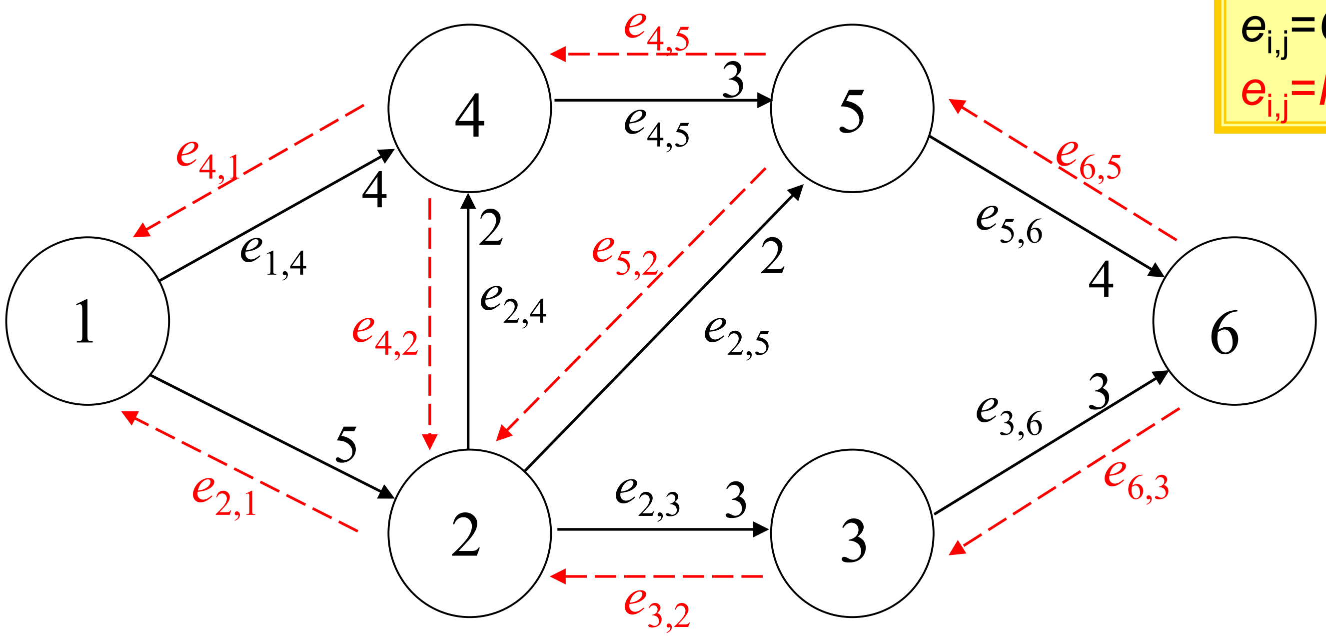
如何设计该方法的实现算法？

```
FORD-FULKERSON(G, s, t)
for each edge (u, v) ∈ G.E
    (u, v).f = 0
while there exists a path p from s to t in the residual network G_f
    c_f(p) = min{c_f(u, v) : (u, v) is in p}
    for each edge (u, v) in p
        if (u, v) ∈ E
            (u, v).f = (u, v).f + c_f(p)
        else (v, u).f = (v, u).f − c_f(p)
```

算法的正确性已经证明，但其效率取决于s-t路径的探知

Labeling Algorithm (Ford & Fulkson)

General Scenario:



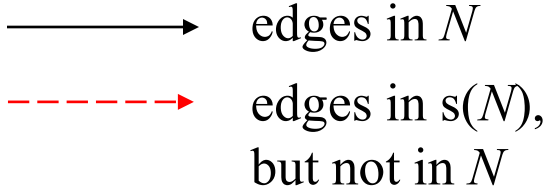
Residual capacity:

$e_{i,j}=C_{i,j}-F_{i,j}$

$e_{i,j}=F_{j,i}$ if $F_{j,i}>0$

$C_{i,j}$ is the capacity of edge (i,j)

$F_{i,j}$ is the flow on edge (i,j)



Labeling Algorithm (Ford & Fulkson)

Initialization: set all flow to 0

Step 1: (1) Identify N1

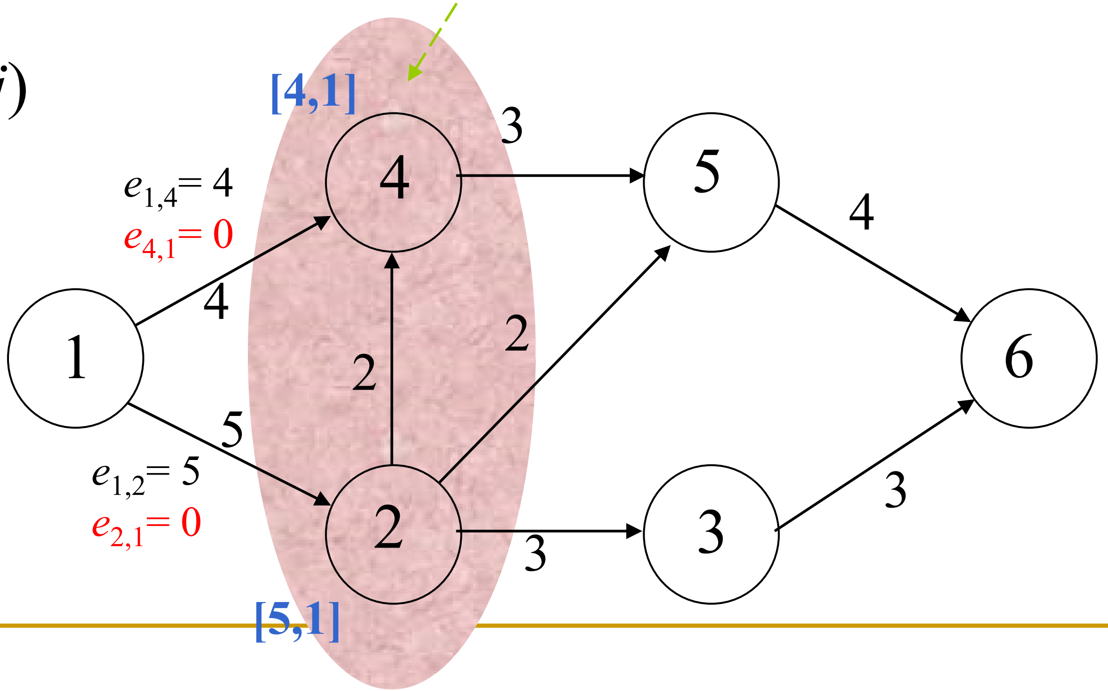
(2) Label nodes in $N1$ as follows

residual capacity of $(1,j)$

$[E_j, 1]$

Denoting node 1

N_1, all nodes connected to the source by an edge with positive residual capacity



Labeling Algorithm (Ford & Fulkson)

Step 2: (1) Identify $N_2(j)$, based on the node j, with the smallest number, in N_1

(2) Label nodes in N2(j) as follows

min$\{E_j$, residual capacity of $(j,k)\}$

$[E_k, j]$

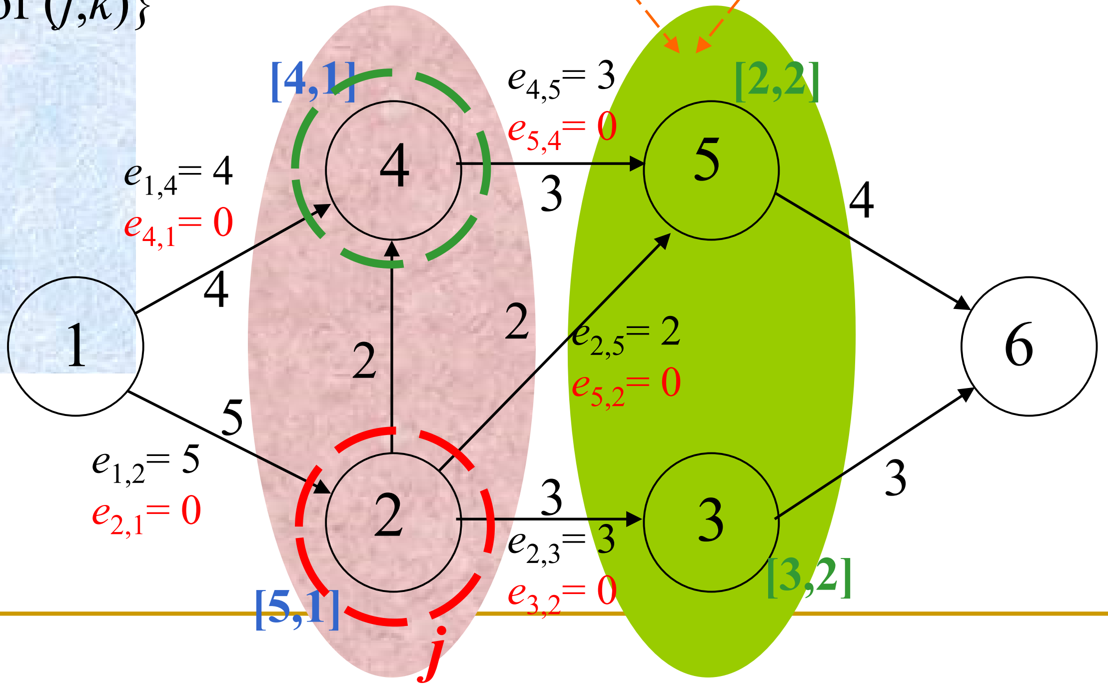
Denoting node j

(3) Do as above for all j in N_1, and let

$$N_2 = \cup_{j \in N_1} N_2(j)$$

$N_2(j)$, all unlabelled nodes connected to node j by an edge with positive residual capacity

Also $N_2(4)$, in this case



Labeling Algorithm (Ford & Fulkson)

Step 3: Continue as in step 2, forming N_3, N_4, N_5, ..., until:

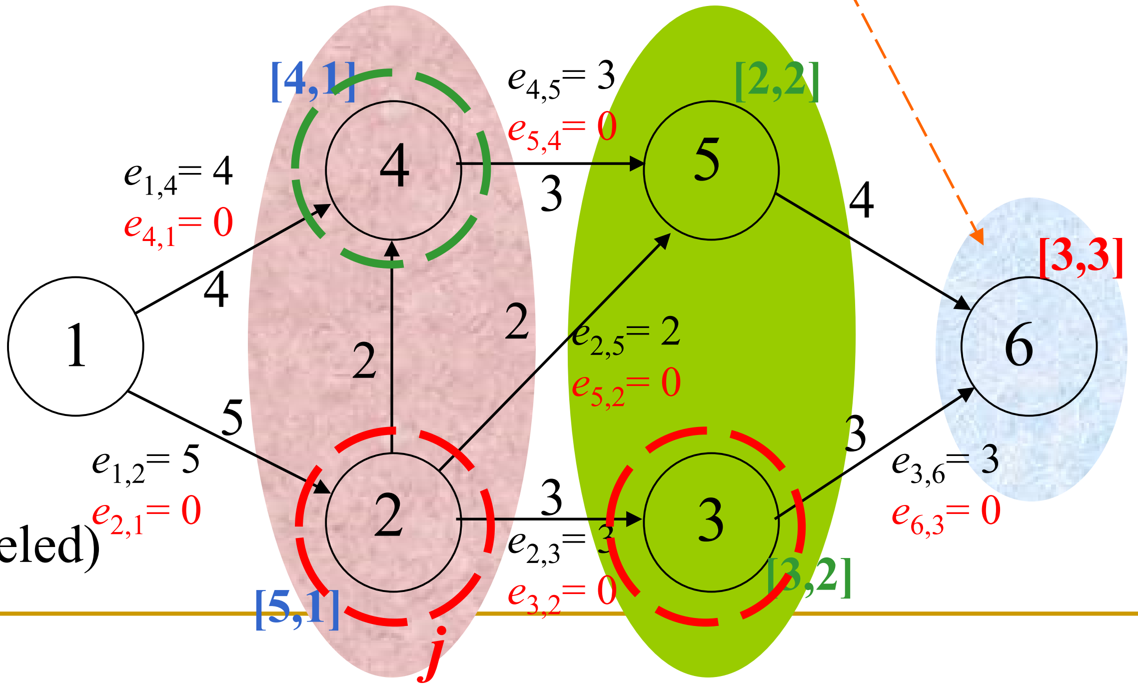
(i) the sink has been labeled, and the total flow is the maximum flow (step 4)

or

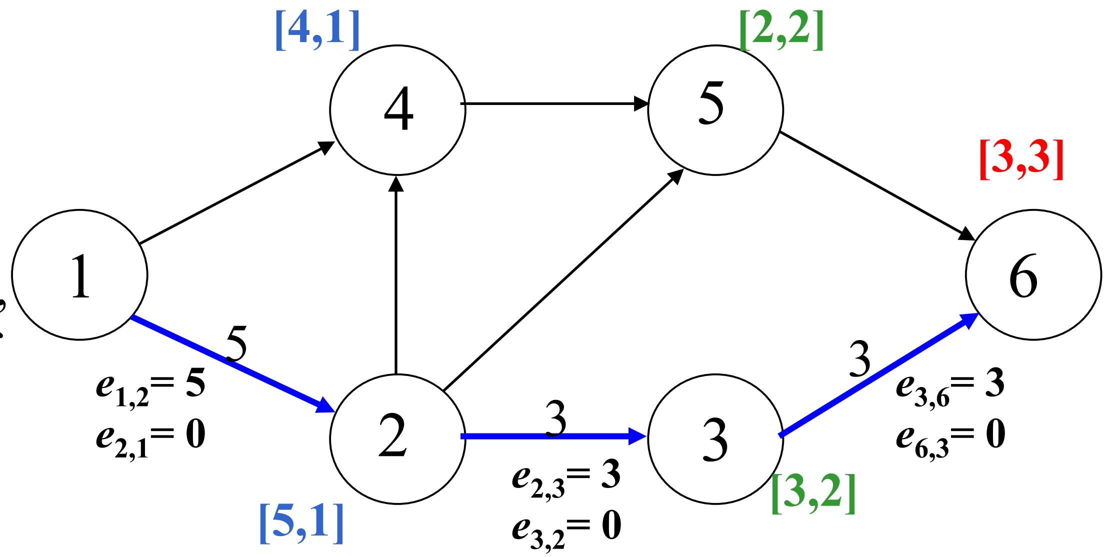
(ii) the sink has not been labeled, but no other nodes can be labeled according to the rules

(note: the source is not labeled)

N_3, the sink has been labeled



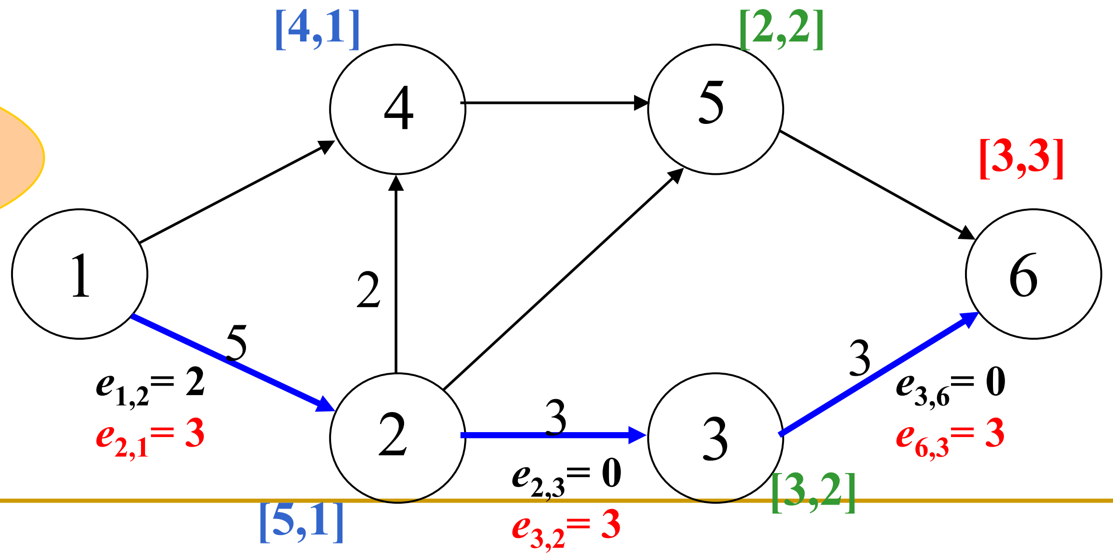
Situation after one full cycle:



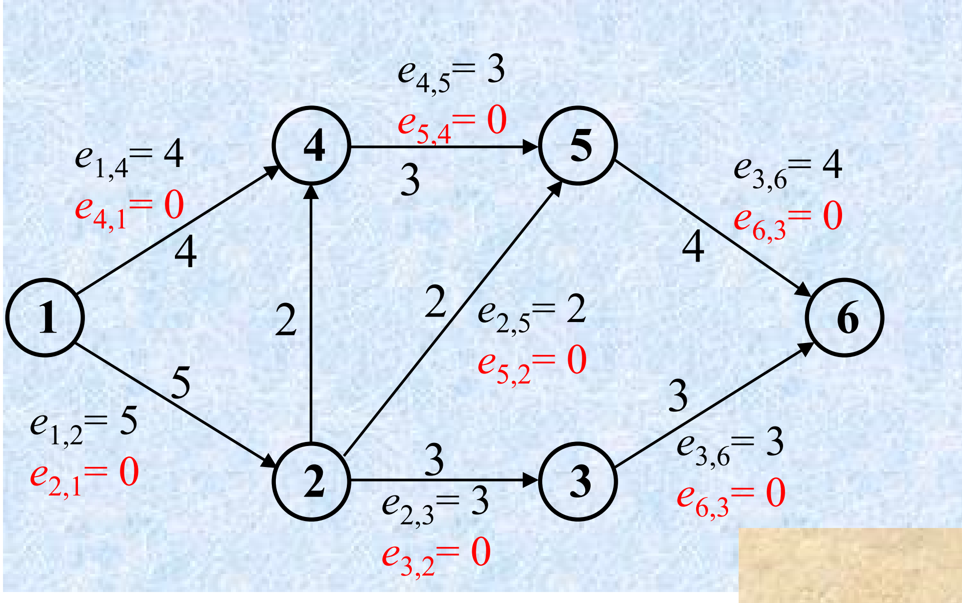
The label of sink is [E_n, m] (here, [3,3]), where E_n is the amount of extra flow that can be made to reach the sink through a path π, and the path can be traced backward by node m

$e_{i,j}$, $e_{j,i}$ are changed accordingly

and then return to step 1

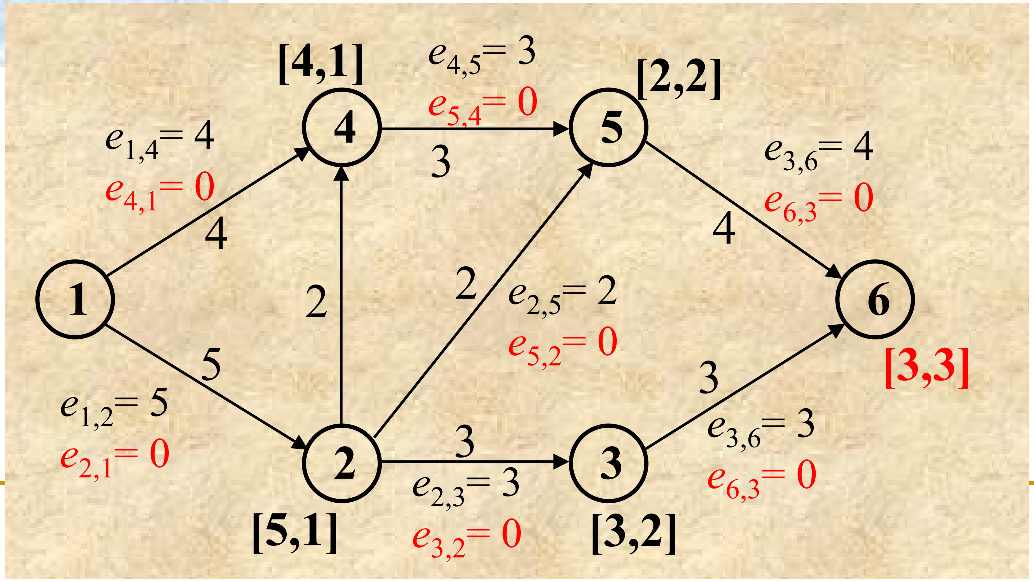


Applying Labeling Algorithm

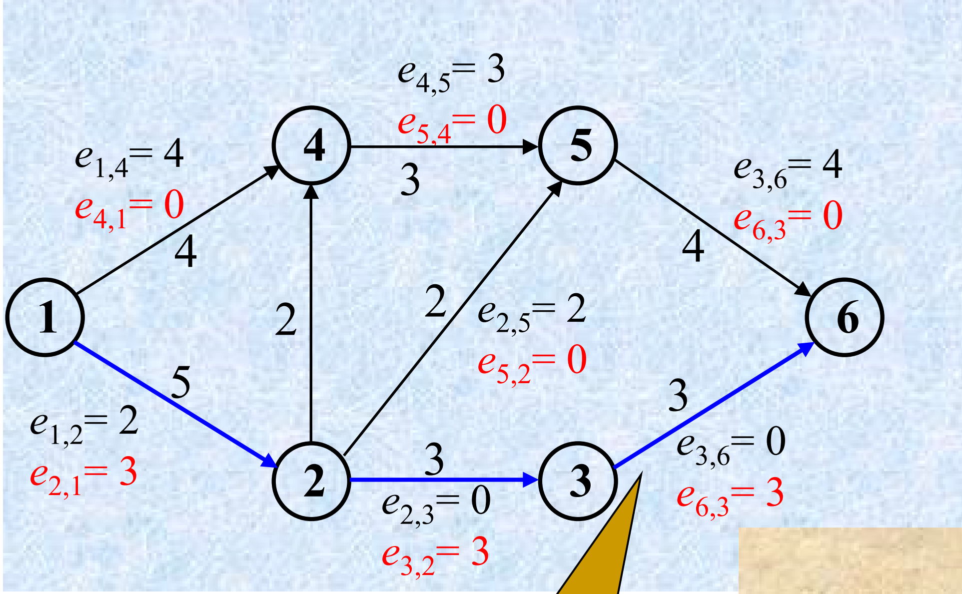


At the beginning, setting all flow to 0

After the first cycle



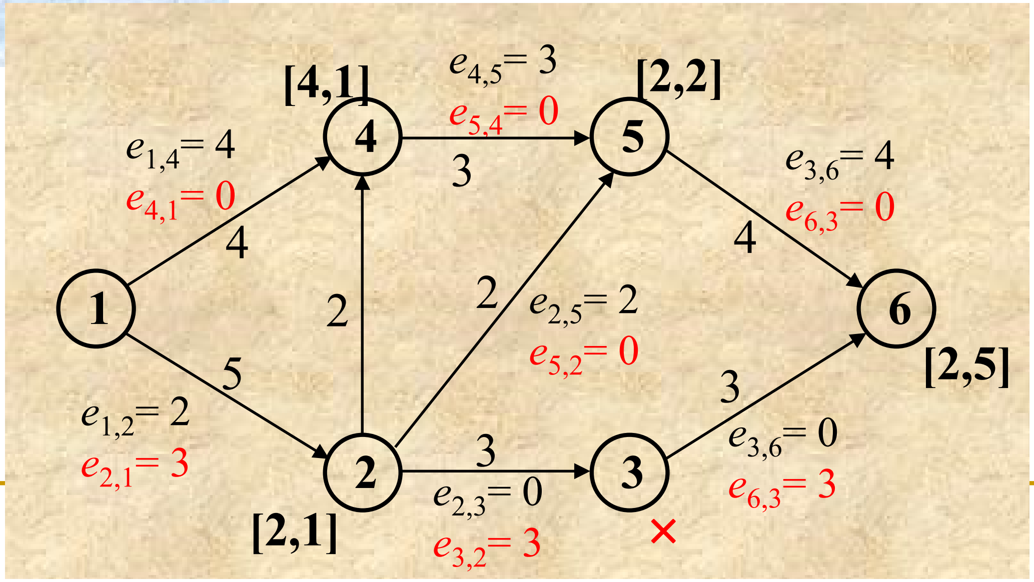
Applying Labeling Algorithm



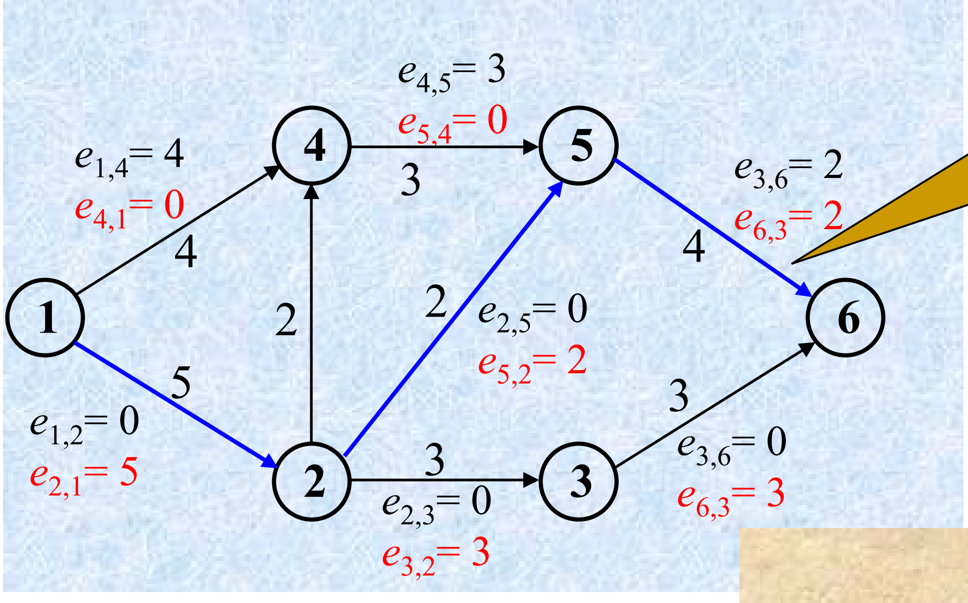
After the first cycle

We increased the flow of this path by 3

After the second cycle



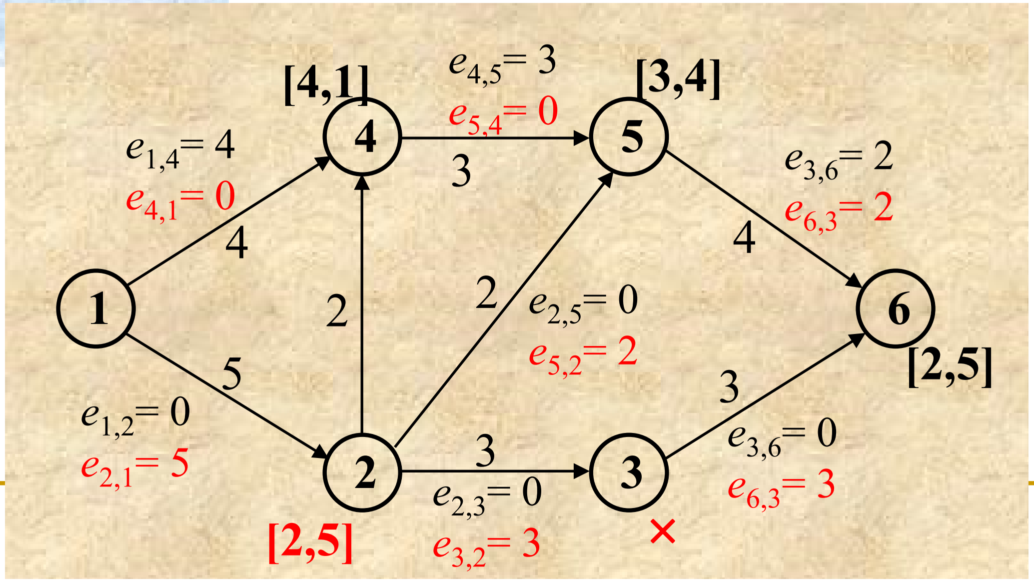
Applying Labeling Algorithm



We increased the flow of this path by 2

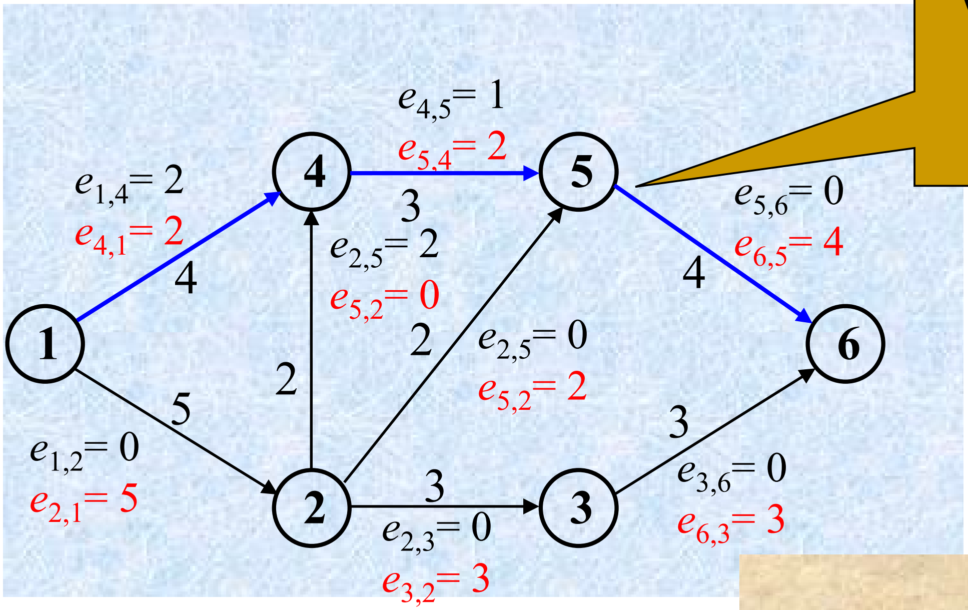
After the second cycle

After the third cycle



Applying Labeling Algorithm

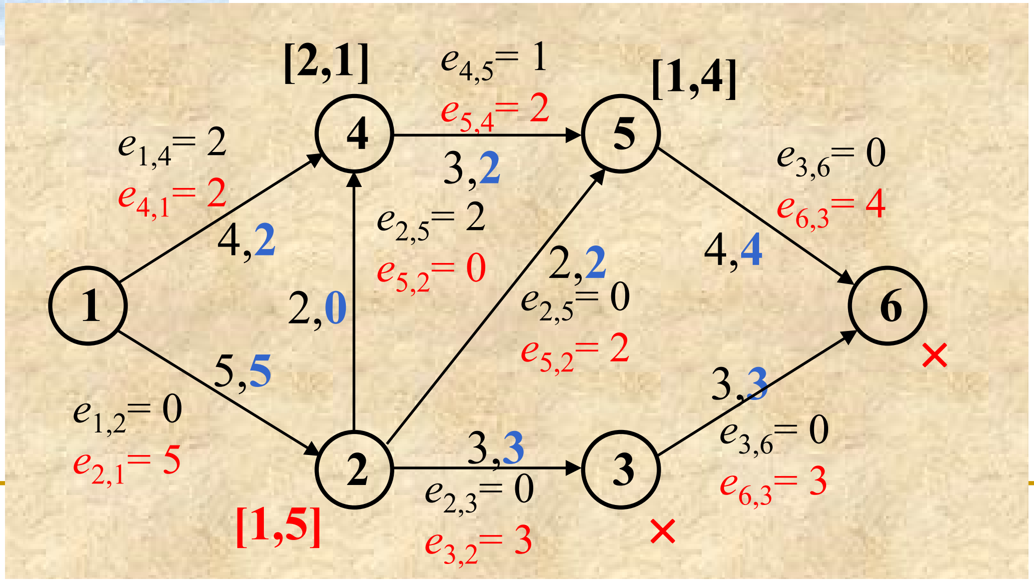
We increased the flow of this path by 2



After the third cycle

After the fourth cycle

The sink has not been labeled, so the final result reached



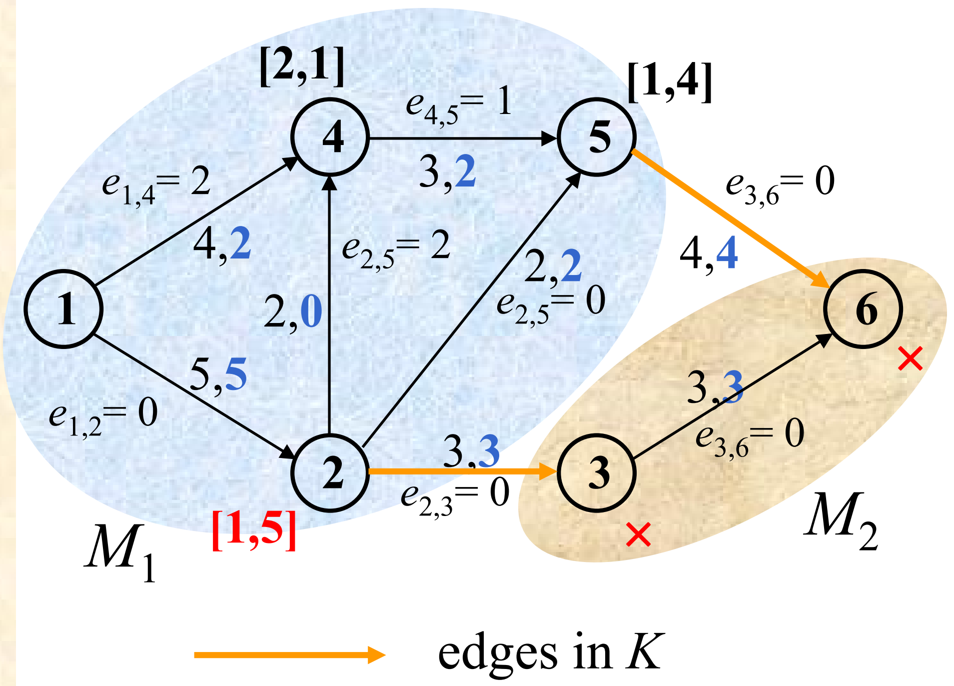
Correctness of Labeling Algorithm

Any path π in N from the source to the sink begins with a node in M_1 and ends with a node in M_2.

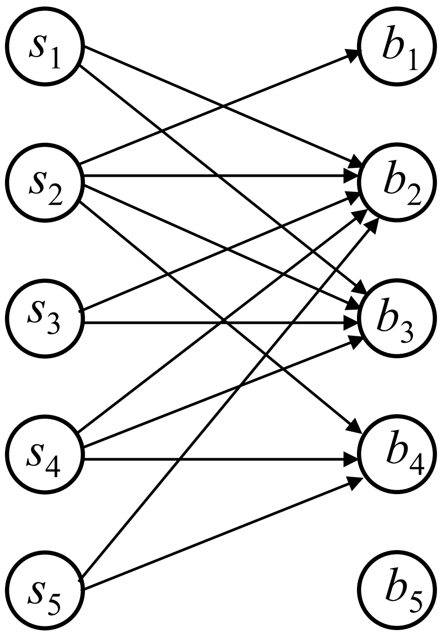
Let K consists of all edges in N that connect a node in M_1 with a node in M_2. So, there must be a edge (i,j), with i is the last node in π that belongs to M_1, and j in M_2. So, (i,j) is in K, and K is a cut.

For all such (i,j), the final flow produced by the algorithm must result in (i,j) carrying its full capacity, otherwise, the positive excess capacity will cause j labeled, contradiction.

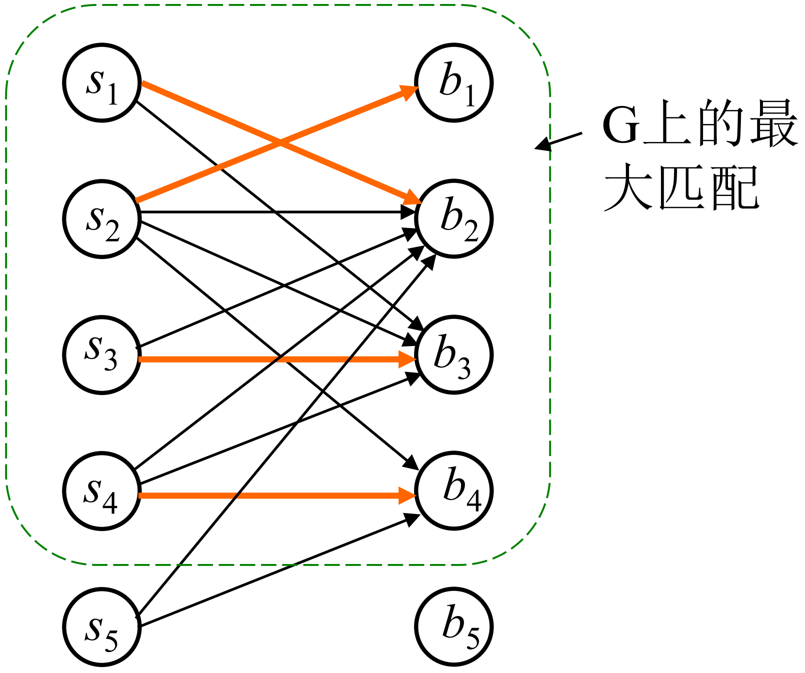
Algorithm stops at Step 4



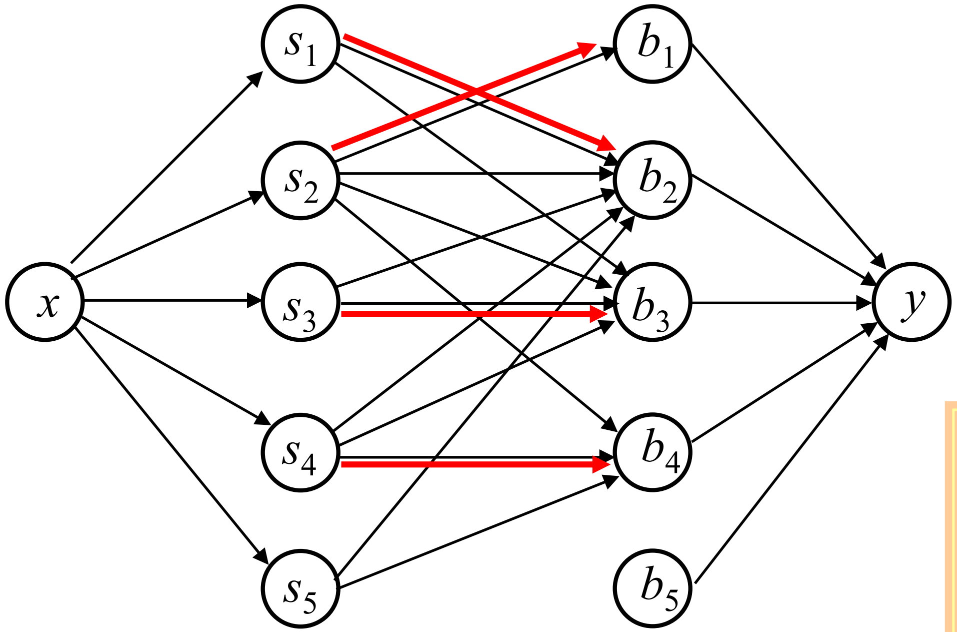
Matching



图G



用网络流来解两步图最大匹配问题



with each capacity set to 1

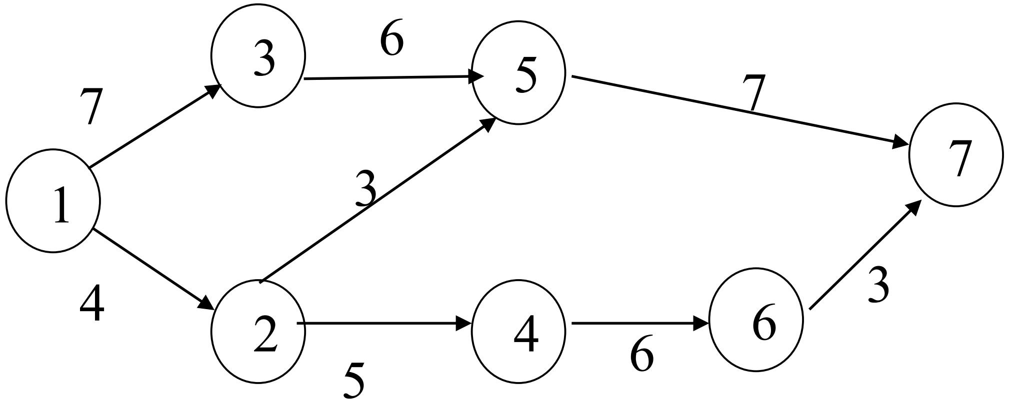
Labeling algorithm for max flow is used in the network to compute the matching

Hall's Marriage Theorem

- Let R be a relation from A to B. Then there exists a complete matching M if and only if for each $X \subseteq A$, $|X| \leq |R(X)|$

Open topics:

1,写出标号算法。用标号法求解以下流网络的最大流



2,利用最大流算法，证明hall定理

课外作业

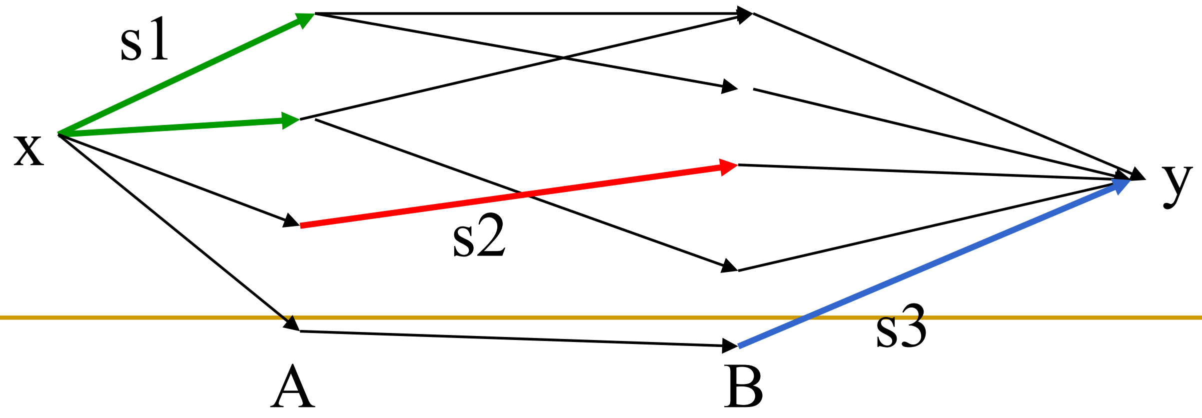
- TC Ex.26.1: 1, 2, 6, 7
- TC Ex.26.2: 2, 6, 8, 10, 12, 13
- TC Ex.26.3: 3
- TC Prob.26: 1, 2

Hall's Marriage Theorem

- Let R be a relation from A to B. Then there exists a complete matching M if and only if for each $X \subseteq A$, $|X| \leq |R(X)|$
- Proof:
  - $\Rightarrow$ Obviously
  - $\Leftarrow$ $|A| = n$
    - add supersource and supersink, if max flow value is $|A|$, then we get it.
    - If min cut has value $|A|$, we get it.

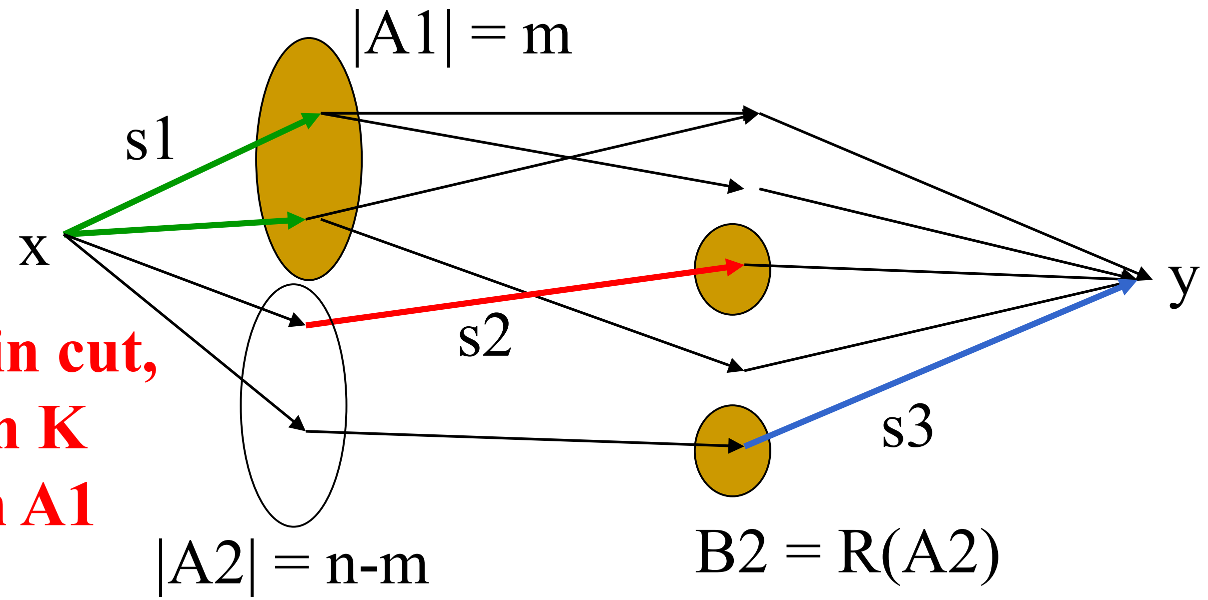
Hall's Marriage Theorem

- Suppose *K* is a minimal cut.
  - We can consider all edges in *K* as in three sets:
    - S_1: those begin at supersource; |s1|=m
    - S_2: those correspond to pairs in *R*;
    - S_3: those end at supersink.



Hall's Marriage Theorem

- Remove S1:



K is the min cut,
No edges in K
begin from A1

|B2| >= |A2| = n-m

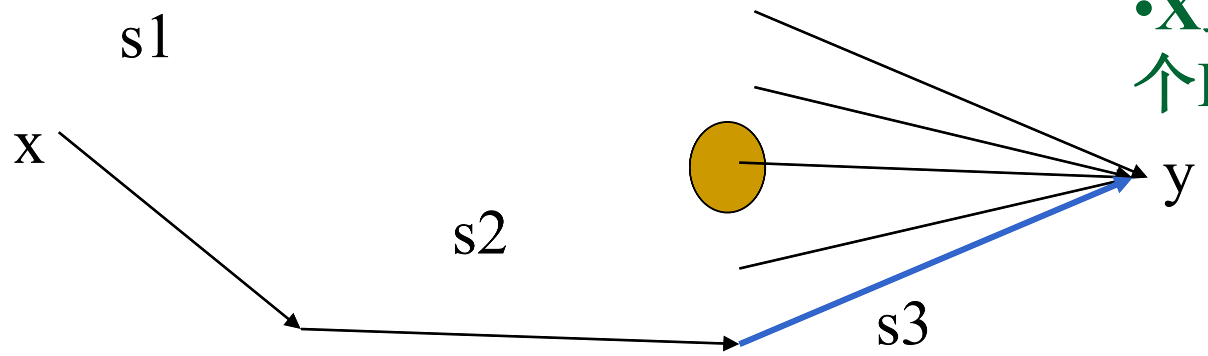
If m = n, s1=n, we get it

precondition

X通过A2集合元素至少和n-m个B2元素相连

Hall's Marriage Theorem

- Remove S2: suppose |S2| = r



•|S1|=m
•X至少和n-m
个B2元素相连

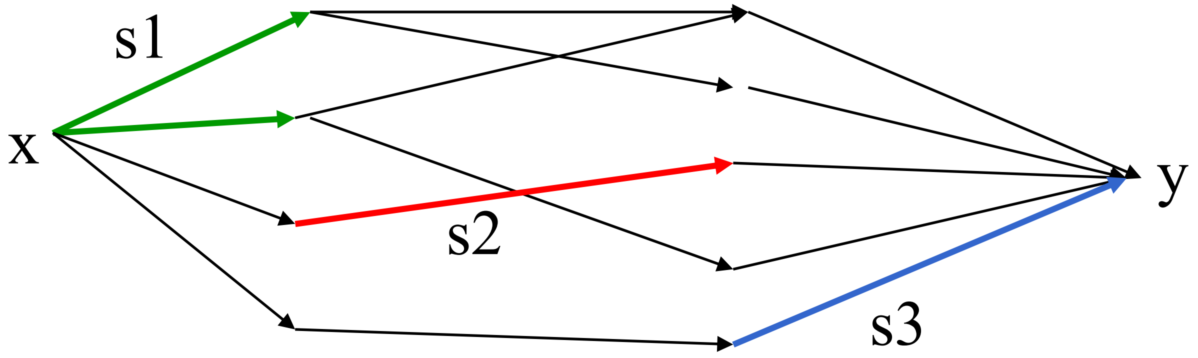
if |S3|=0, |S2| = n-m, |K|= |S1|+|S2| = n

X is still connected to at least (n-m)-r elements of B

if |S3|>0

Hall's Marriage Theorem

- K is a min cut:
  - X is still connected to at least (n-m)-r elements of B
  - |S3| >= (n-m)-r



$|K| = |S1|+|S2|+|S3| >= m + r+ n\text{-}m\text{-}r = n$

Proof of Hall's Theorem:

|K| >= n, So, the following greens is one of the min cut.
So , the following reds is one of the complete matching