Transport networks （传输网）

Zhang Yanmei

ymzhang@bupt.edu.cn

College of Computer Science & Technology

Beijing University of Posts & Telecommunications

Content

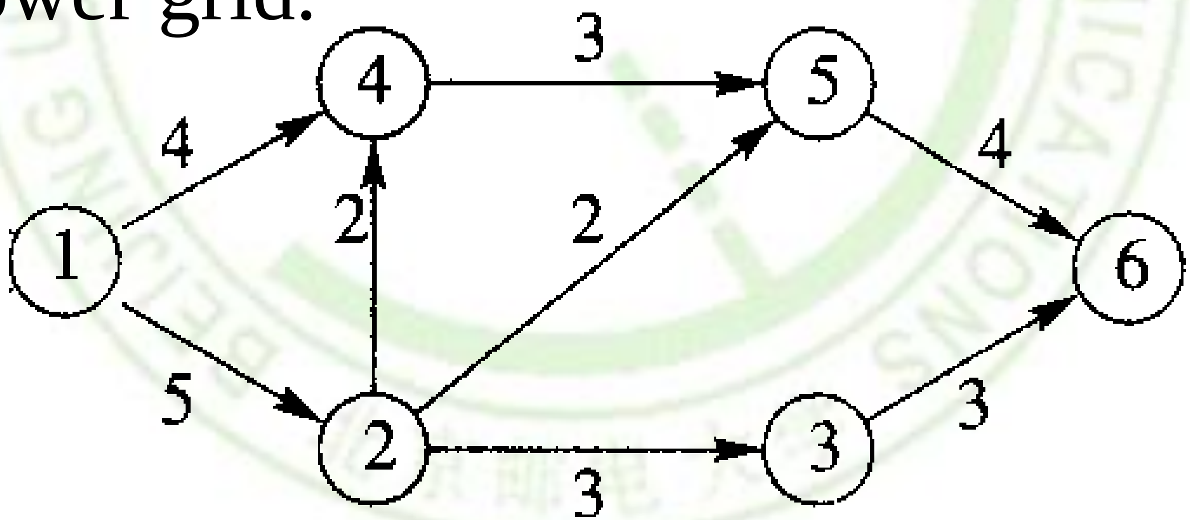
- Application of digraph
- Concept
  - Capacity (容量)
  - Transport network (传输网)
  - Flow (流量)
  - Virtual flow (虚流量)
  - Maximal flow (最大流量)
- Labeling algorithm　(标记算法)

Directed graph (Digraph)

- An important use of labeled digraphs is to model what are commonly called transport networks.
- The label on an edge represents the maximum flow that can be passed through that edge and is called the *capacity*（容量） of the edge. Many situations can be modeled in this way.

Example

- Figure below might as easily represent an oil pipeline, a highway system, a communications network, or an electric power grid.



Example

- The vertices of a network are usually called nodes and may denote pumping stations, shipping depots, relay stations, or highway interchanges.

Transport network

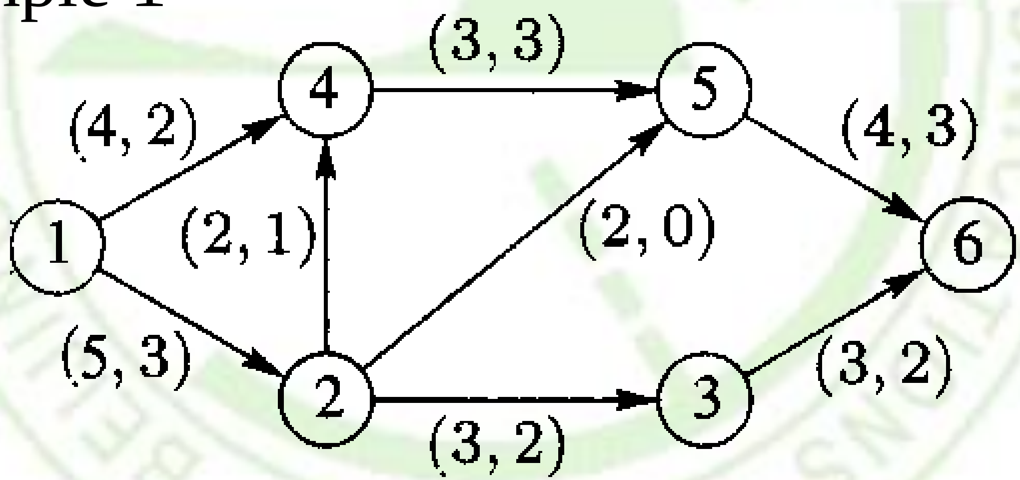
- More formally, a *transport network*, or a *network*, is a connected digraph N with the following properties:
  - There is a unique node, the *source*（*源端*）, that has in-degree 0. We generally label the source node l.
  - There is a unique node, the *sink*（*宿端*）, that has out-degree 0. If N has n nodes, we generally label the sink as node n.
  - The graph N is labeled. The label, C_{ij}, on edge (i, j) is a nonnegative number called the *capacity* of the edge.

Flows

- Mathematically, a *flow（流量）* in a network N is a function that assigns to each edge (i, j) of N a nonnegative number F_{ij} that does not exceed C_{ij}.
  - *Conservation of flow（流量守恒）*
  - *Value of the flow*

Flows

- We can represent a flow F by labeling each edge (i, j) with the pair (C_{ij}, F_{ij})
- Example 1



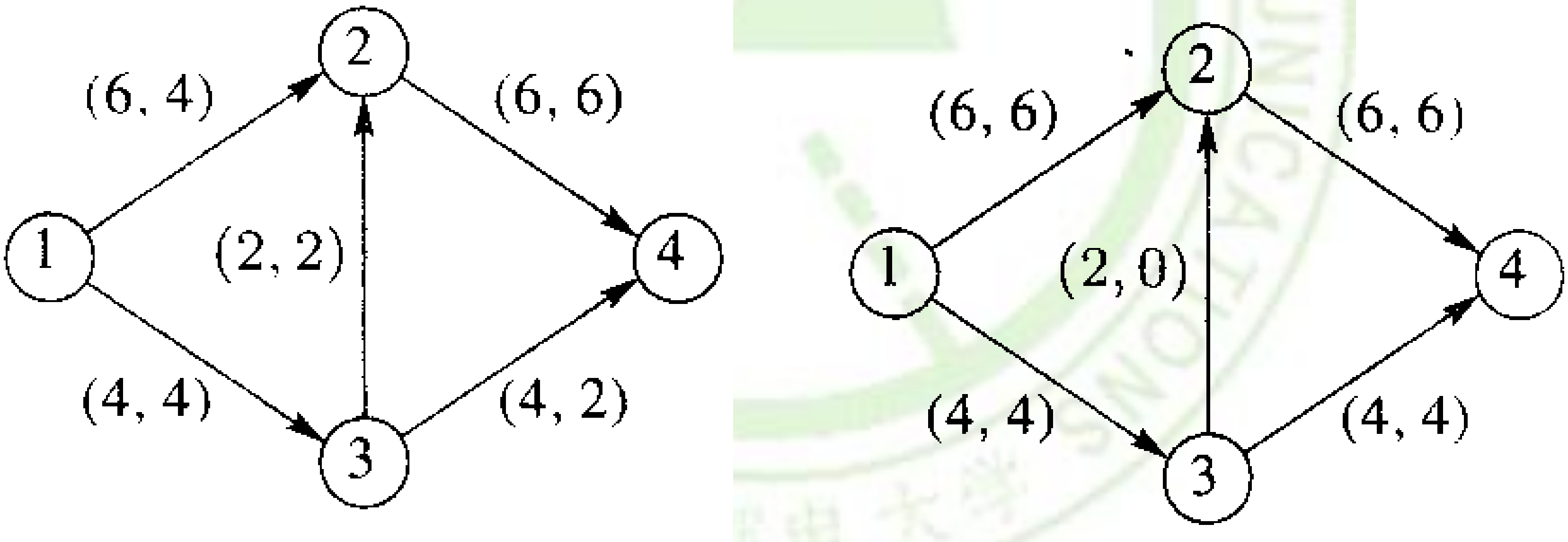
- Here value(F) = 5

Maximum Flows（最大流量）

- For any network an important problem is to determine the maximum value of a flow through the network and to describe a flow that has the maximum value.
- For obvious reasons this is commonly referred to the *maximum flow problem*.

Example 2

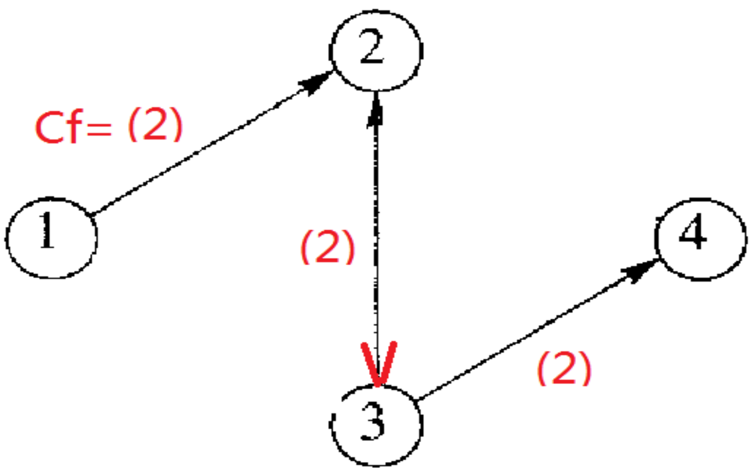
- Even for a small network, we need a systematic procedure for solving the maximum flow problem.



(a) (b)

augment path included virtual flow

An augmenting path is a path $(u_1, u_2, \ldots, u_k)$ in the residual network, where $u_1 = s$, $u_k = t$, and $c_f(u_i, u_{i+1}) > 0$. A network is at maximum flow if and only if there is no augmenting path in the residual network.



Algorithm Ford -Fulkerson

- Inputs Graph G with flow capacity c, a source node s, and a sink node t
- Output A flow f from s to t which is a maximum

- 1. $f(u,v) \leftarrow 0$ for all edges (u,v)
- 2. While there is a path p from s to t in G_f, such that $c_f(u,v) > 0$ for all edges $(u,v) \in p$:
  - 1. Find $c_f(p) = \min\{c_f(u,v) | (u,v) \in p\}$
  - 2. For each edge $(u,v) \in p$
    - 1. $f(u,v) \leftarrow f(u,v) + c_f(p)$ (Send flow along the path)
    - 2. $f(v,u) \leftarrow f(v,u) - c_f(p)$ (The flow might be "returned" later)

A Maximum Flow Algorithm

- The algorithm we present is due to Ford and Fulkerson and is often called the *labeling algorithm（标记算法）*.
- The labeling referred to is an additional labeling of nodes.
- We have used integer capacities for simplicity, but Ford and Fulkerson show that this algorithm will stop in a finite number of steps if the capacities are rational numbers.

residual capacity

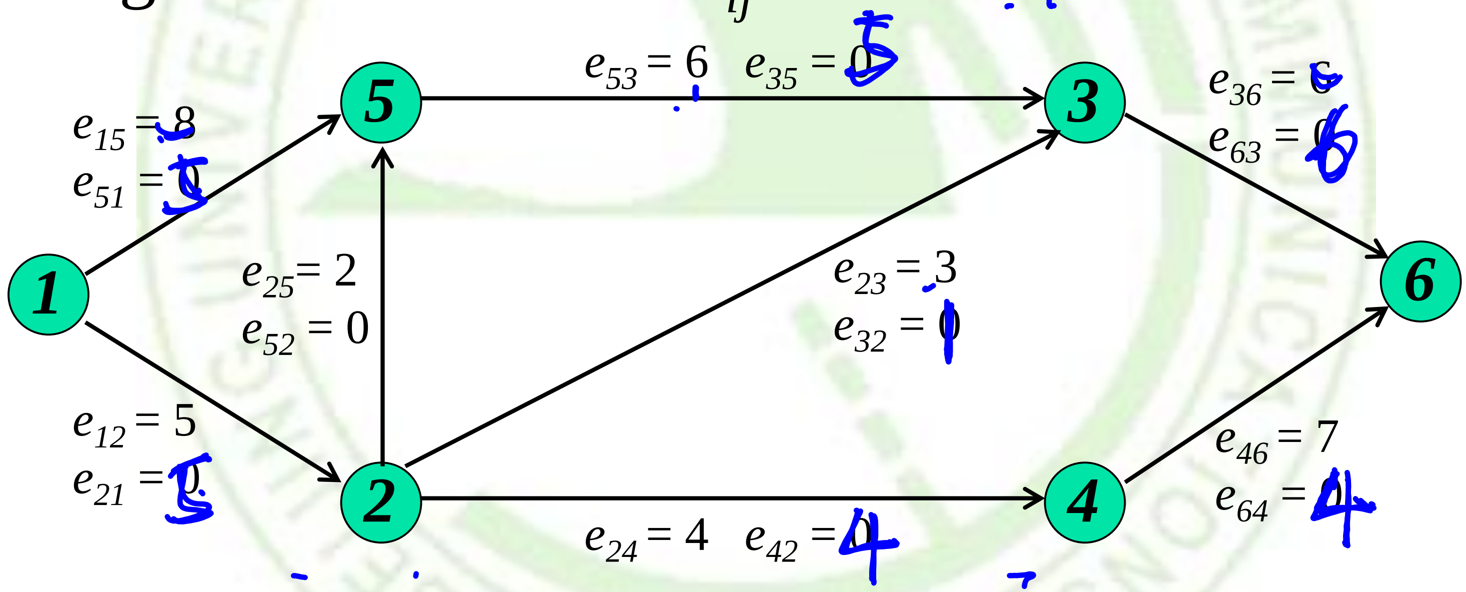
- Let N be a network and let G be the symmetric closure of N.
  - Choose a path in G and let an edge (i,j) in this path.
  - If $(i,j)\in N$ and $e_{ij}=C_{ij}-F_{ij}>0$,then we say this edge has positive excess capacity（剩余容量）.
  - If (i,j) is not an edge of N then we are traveling this edge in the wrong direction. In this case $e_{ij}=F_{ji}$ if $F_{ji}>0$. Increasing flow through edge(i,j) will have the effect of reducing F_{ji}.

Begin

- Begin with all flows set to 0.
- but all edges labeled as residual capacity（剩余容量）.
  - If $(i,j) \in N$, then $e_{ij}=C_{ij}-F_{ij}=C_{ij}-0=C_{ij}$.
  - If (i,j) is not an edge of N, then $e_{ij}=F_{ji}=0$.

Example 4

- The initial flow in all edges is zero. That mean all edges is labeled 0 or C_{ij}.



Step 1

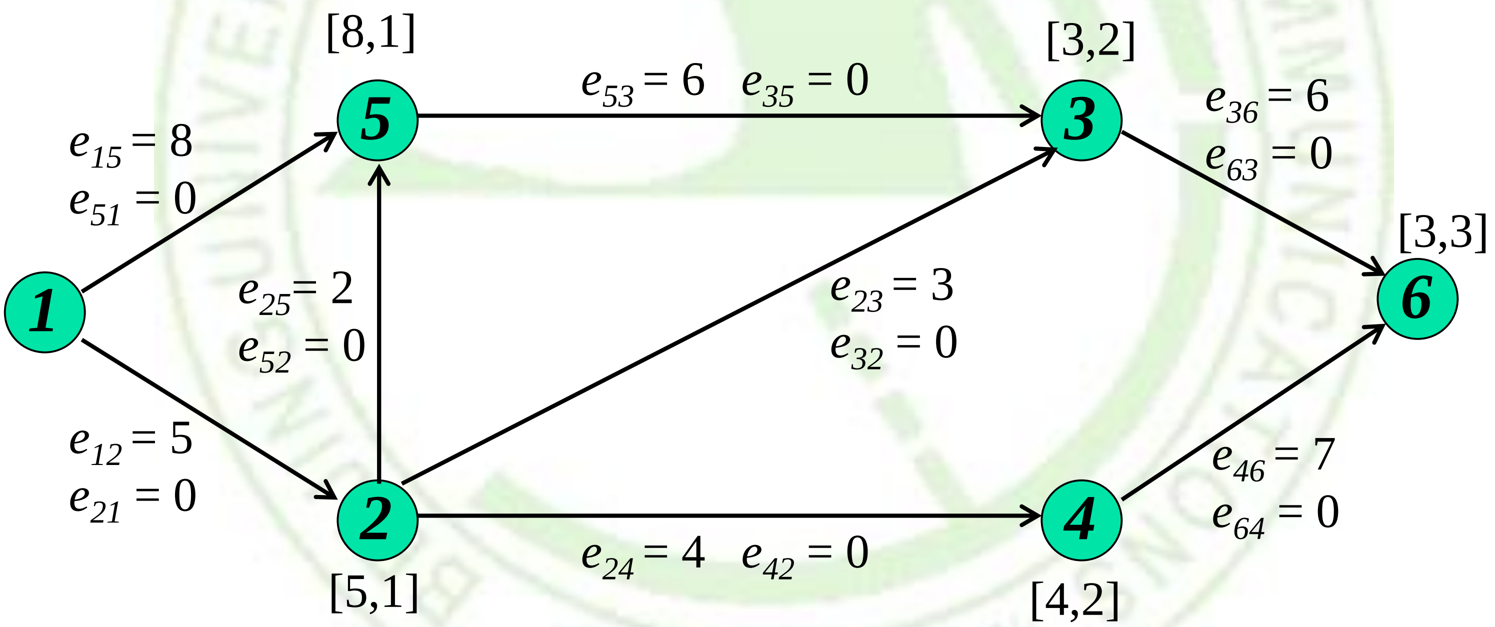
- Let N_1 be the set of all nodes connected to the source by an edge with positive excess capacity.
- Label each j in N_1 with $[E_j, 1]$, where E_j is the excess capacity e_{1j} of edge $(1, j)$.
- The 1 in the label indicates that j is connected to the source, node 1.

Step 2

- Let node j in N_1 be the node with smallest node number and let N_2(j) be the set of all unlabeled nodes, other than the source, that are joined to node j and have positive excess capacity.
- Suppose that node k is in N_2(j) and (j, k) is the edge with positive excess capacity. Label node k with [E_k, j], where E_k is the minimum of E_j and the excess capacity e_{jk} of edge (j, k).
- **When all the nodes in $N_2(j)$ are labeled in this way, repeat this process for the other nodes in N_1. Let** $N_2 = \bigcup_{j \in N_1} N_2(j)$

Step 1,2,3

- $N_1=\{2,5\},N_2=\{3,4\},N_3=\{6\}$



Step 3

- Repeat Step 2, labeling all previously unlabeled nodes N_3 that can be reached from a node in N_2 by an edge having positive excess capacity.
- Continue this process forming sets N_4, N_5. . . . until after a finite number of steps either
  - (i) the sink has not been labeled and no other nodes can be labeled. It can happen that no nodes have been labeled; remember that the source is not labeled. or
  - (ii) the sink has been labeled.

Step 4

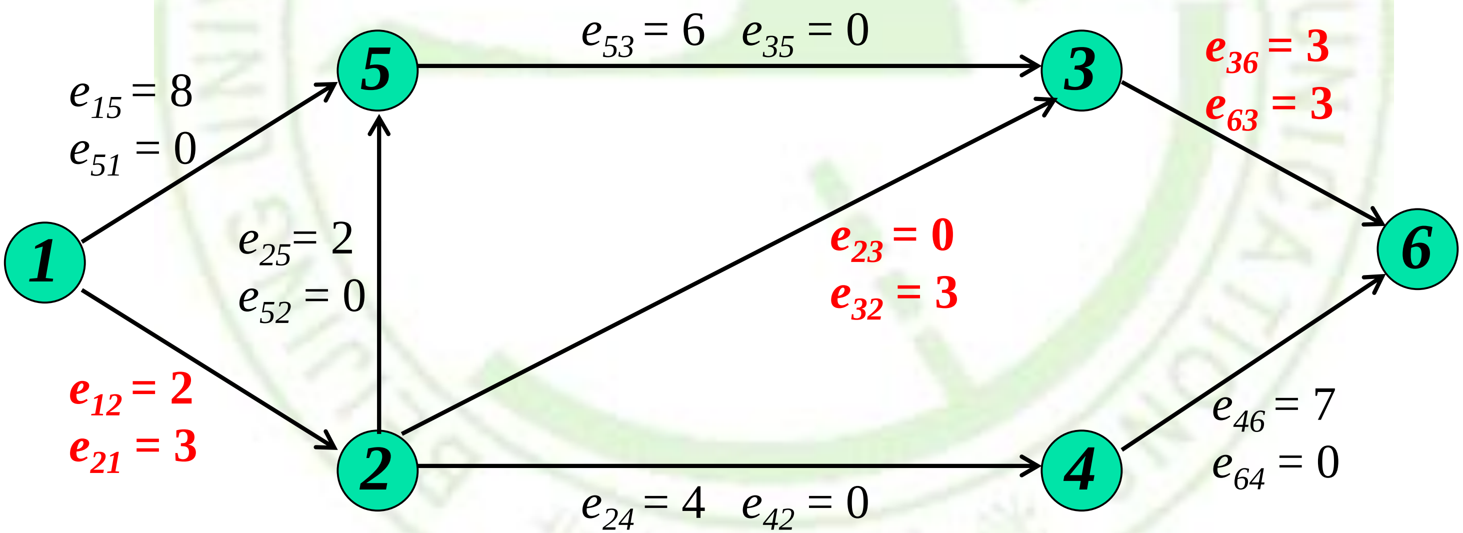
- In case (i), the algorithm terminates and the total flow then is a maximum flow.
  - (i) the sink has not been labeled and no other nodes can be labeled.

Step 5

- In case (ii) the sink, node n, has been labeled with [E_n, m] where E_n is the amount of extra flow that can be made to reach the sink through a path π.
  - (ii) the sink has been labeled.

Step 5

- Path π={1,2,3,6}, E_n=3, all edges in π e_{ij} - E_n , and the symmetric edges e_{ji} + E_n .



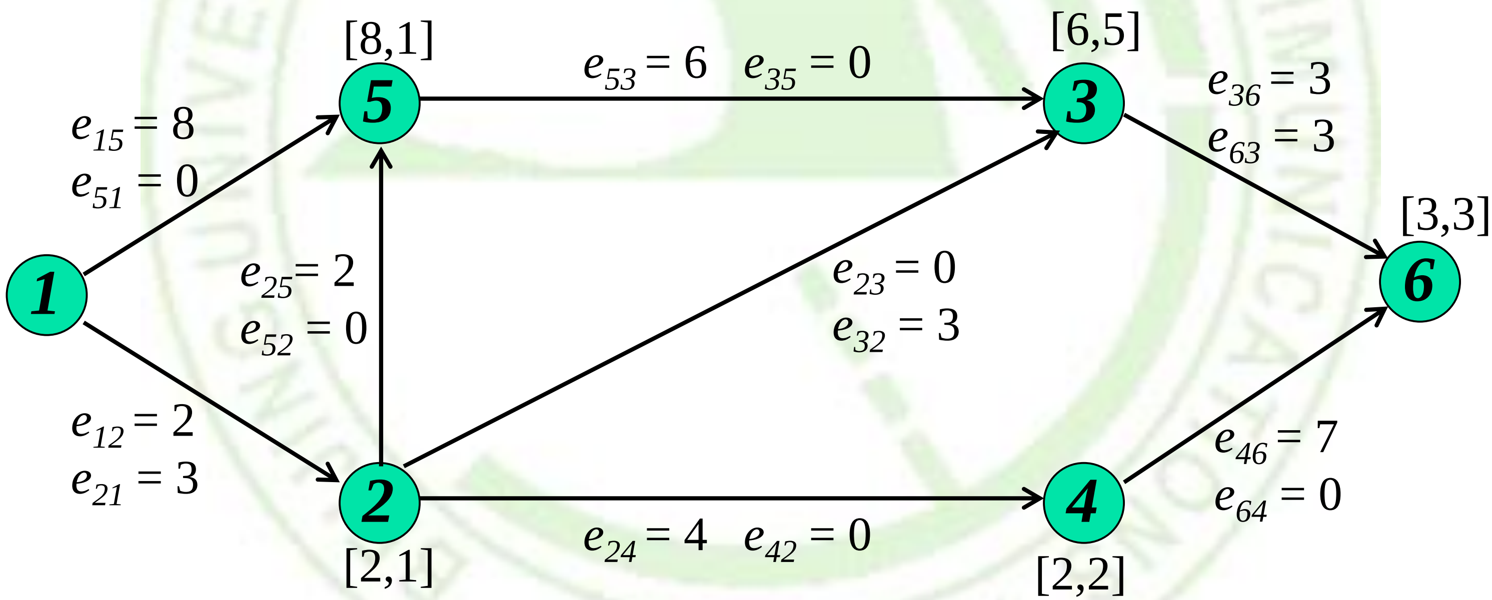
Step 5

- We examine π in reverse order.
  - decrease the excess capacity e_{ij} by E_n .
  - Simultaneously, we increase the excess capacity of the edge (j, i) by E_n.
  - We now have a new flow that is E_n units greater than before and we return to Step 1.

Example 4

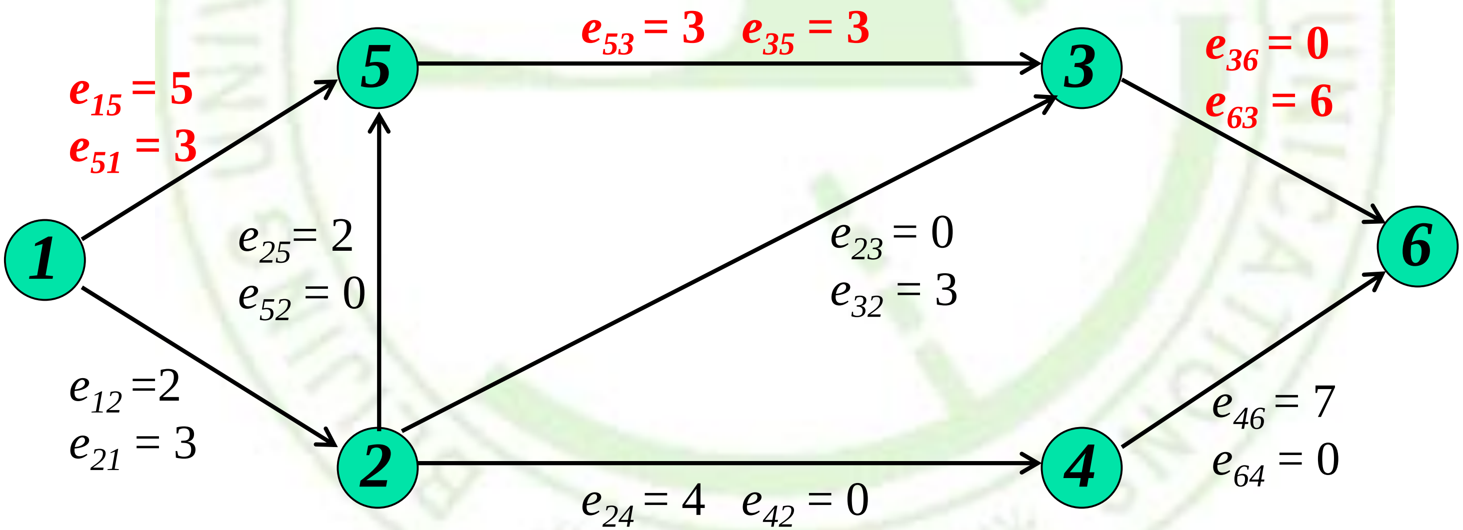
Repeat Step 1,2,3

- $N_1=\{2,5\}$,**$N_2=N_2(2)\cup N_2(5)=\{4,3\}$**,$N_3=\{6\}$



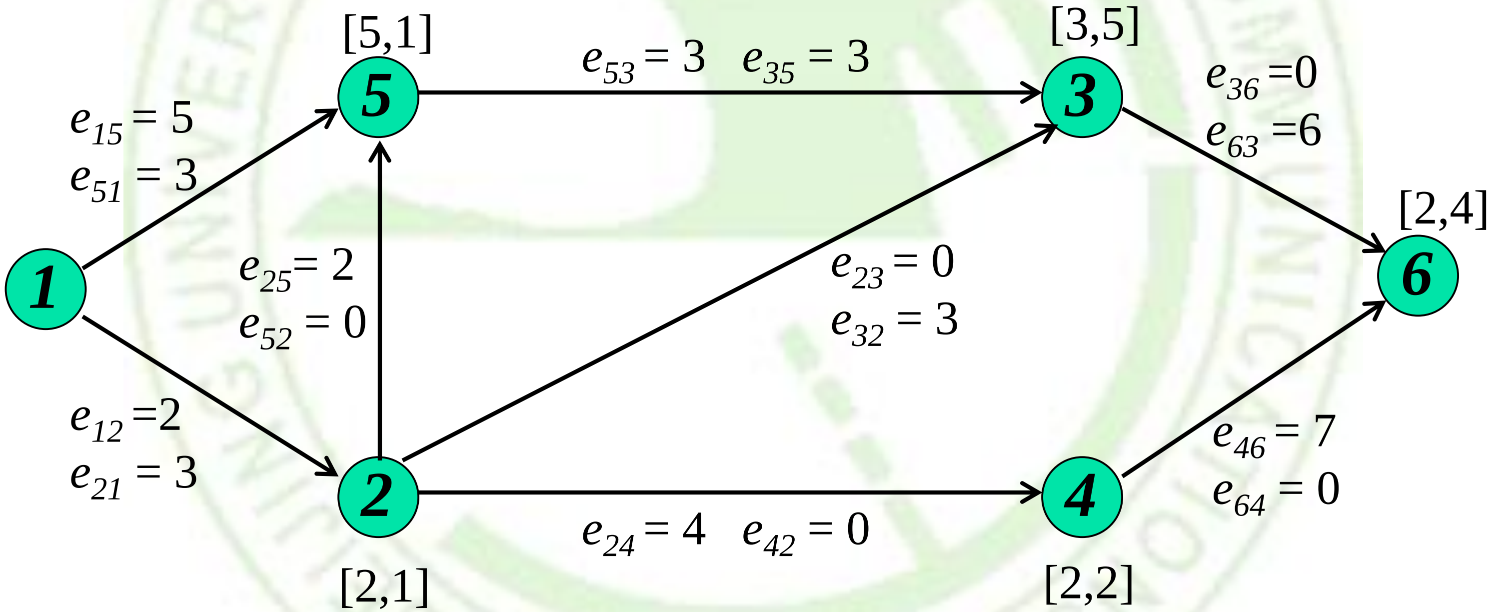
Step 5

- $\pi=\{1,5,3,6\}$, $E_n=3$, all edges in π $e_{ij} - E_n$, and the symmetric edges $e_{ji} + E_n$.



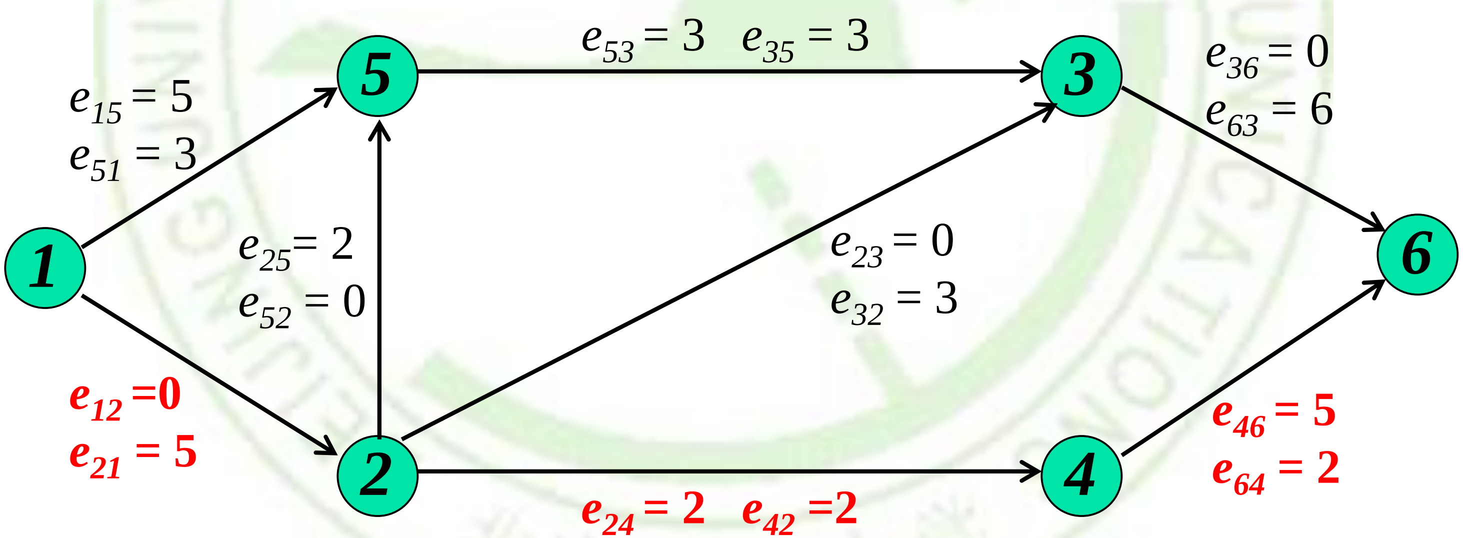
Repeat Step 1,2,3

- $N_1=\{2,5\}$,$N_2=N_2(2)\cup N_2(5)=\{4,3\}$,$N_3=\{6\}$



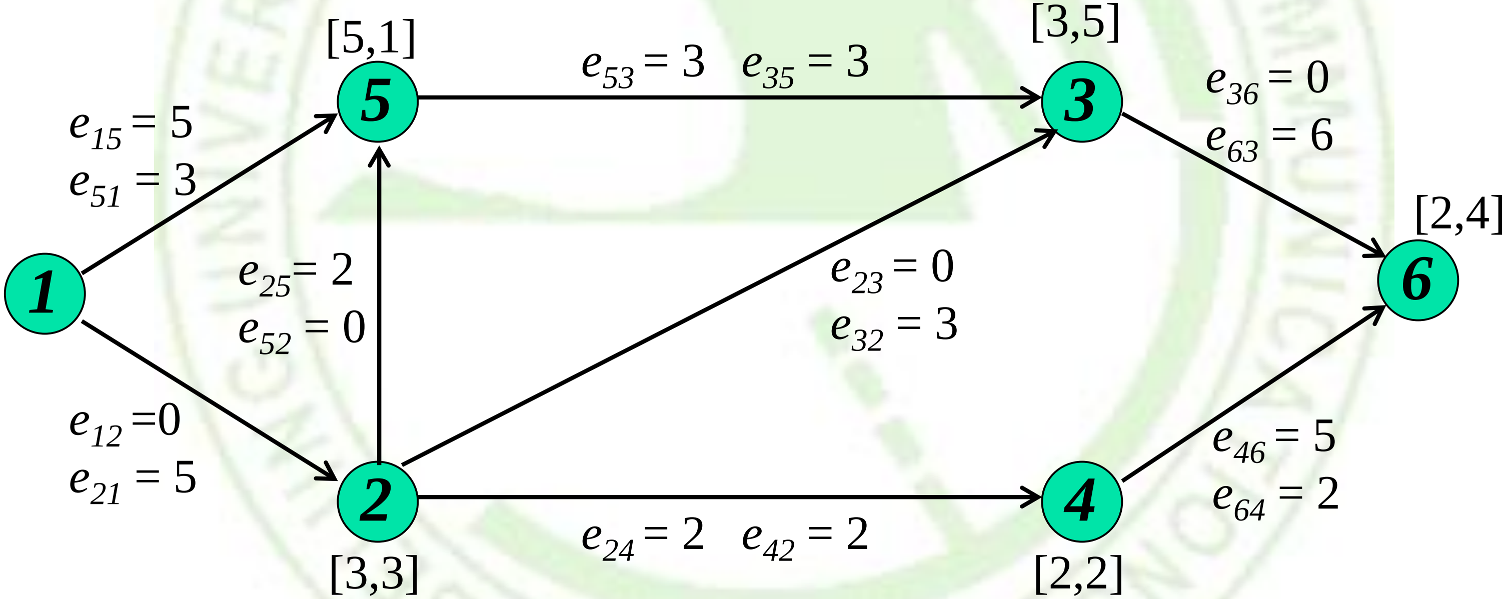
Step 5

- π={1,2,4,6}, E_n =2, all edges in π e_{ij} - E_n , and the symmetric edges e_{ji} + E_n .



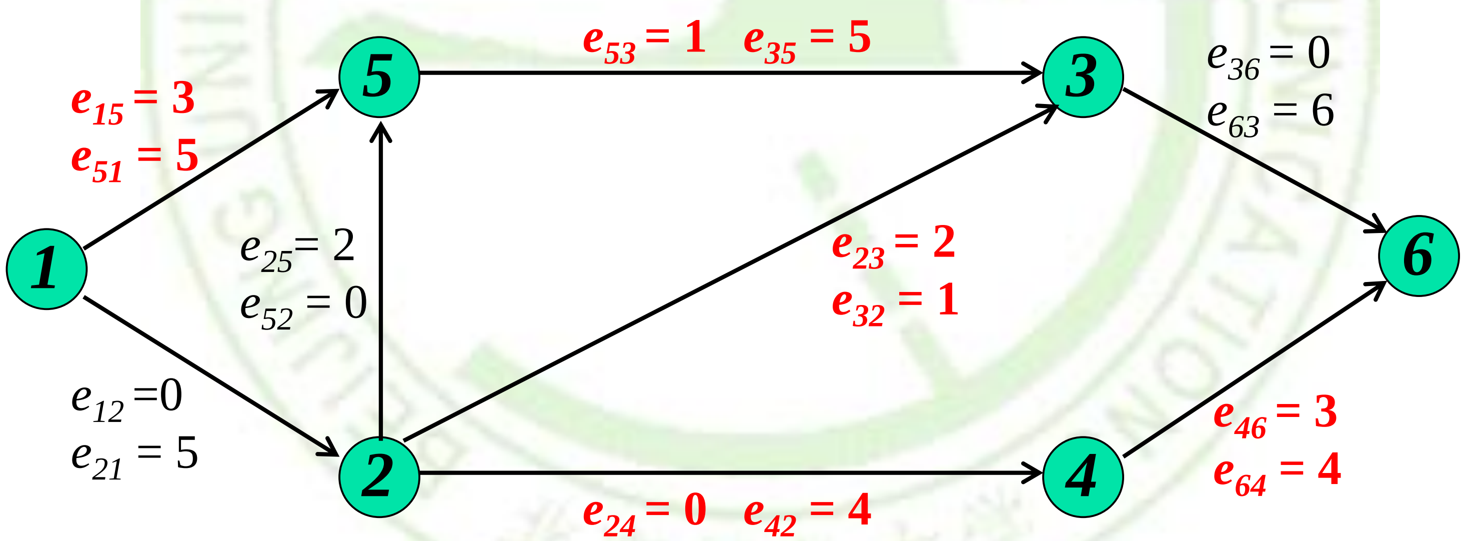
Repeat Step 1,2,3…

- $N_1=\{5\},N_2=\{3\},N_3=\{2\},N_4=\{4\},N_5=\{6\}$



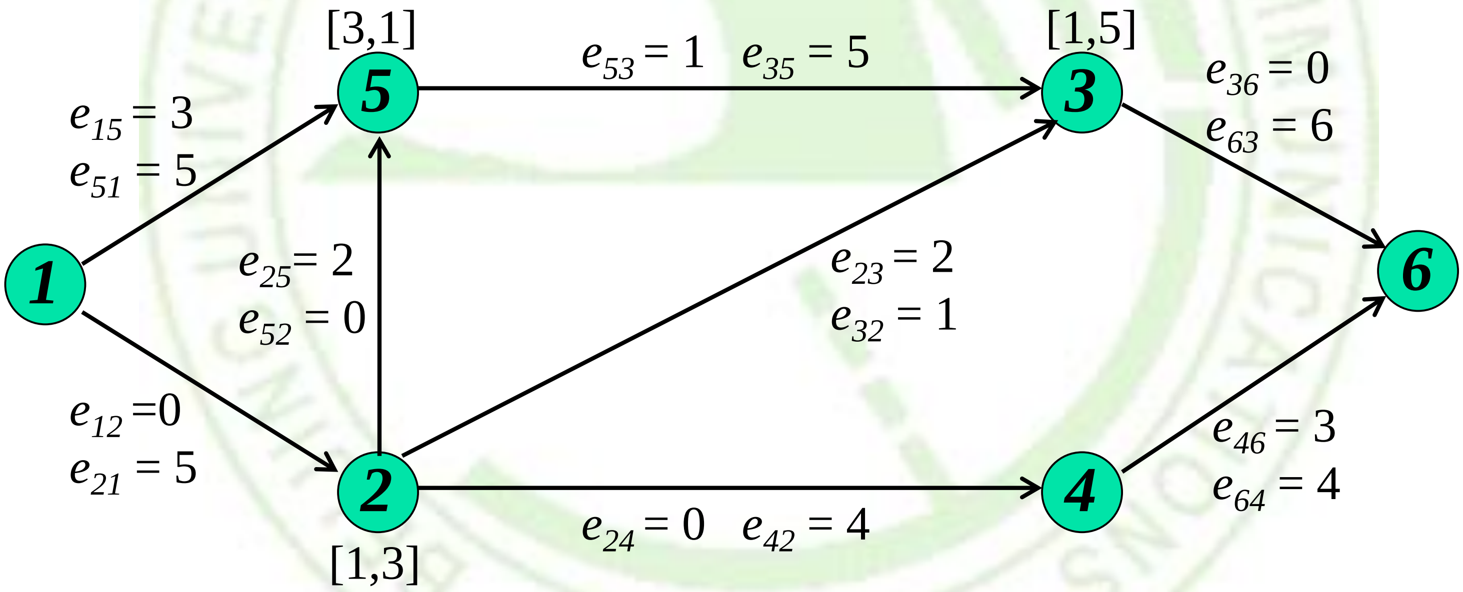
Step 5

- $\pi=\{1,5,3,2,4,6\}$, $E_n=2$, all edges in π $e_{ij} - E_n$, and the symmetric edges $e_{ji} + E_n$.



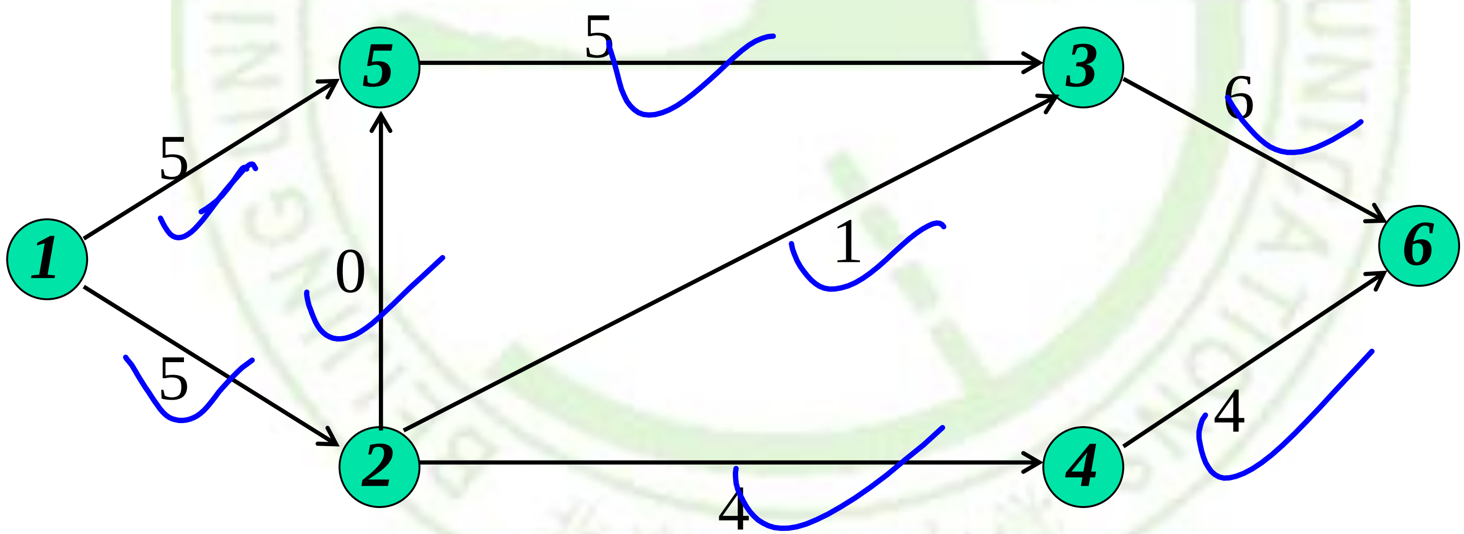
Go further?

- Step 1,2,3 $N_1=\{5\}, N_2=\{3\}, N_3=\{2\}, N_4=\Phi$



Step 4

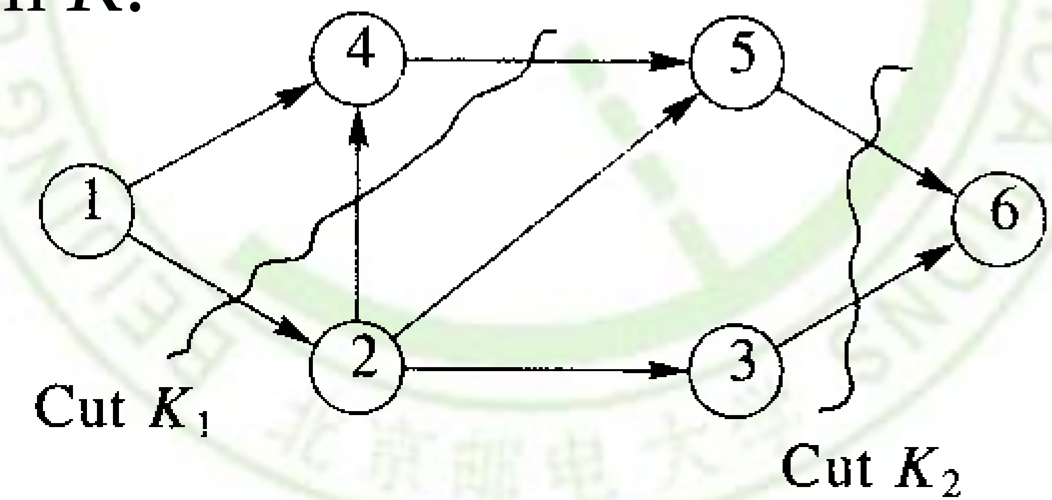
- Terminated with the final overall flow 10. All virtual edges $e_{ji} = F_{ij}$.



正向找完之后看反边出的走哪…

Cut（割集）

- A *cut* in a network N is a set K of edges having the property that every path from the source to the sink contains at least one edge from K.



Capacity of a cut *K*

- The *capacity* of a cut K, $c(K)$, is the sum of the capacities of all edges in K.
- If F is any flow and K is any cut, then
  - $value(F) \le c(K)$.

The Max Flow Min Cut Theorem

- A maximum flow F in a network has value equal to the capacity of a minimum cut of the network.

Find the maximal flow for the network N given in Figure 1.

10

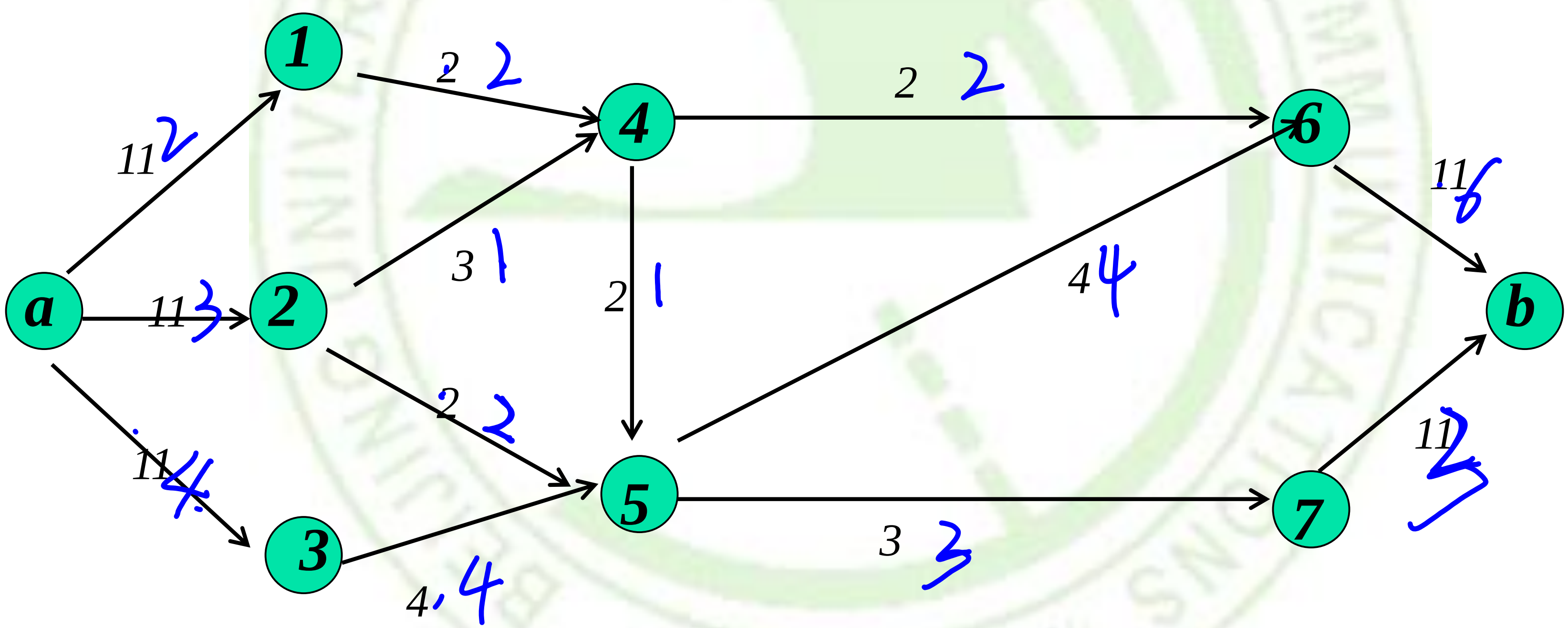


Figure 1

homework

- 10,14@305