Graph Models and Applications

Lecture 11

Discrete Mathematical Structures

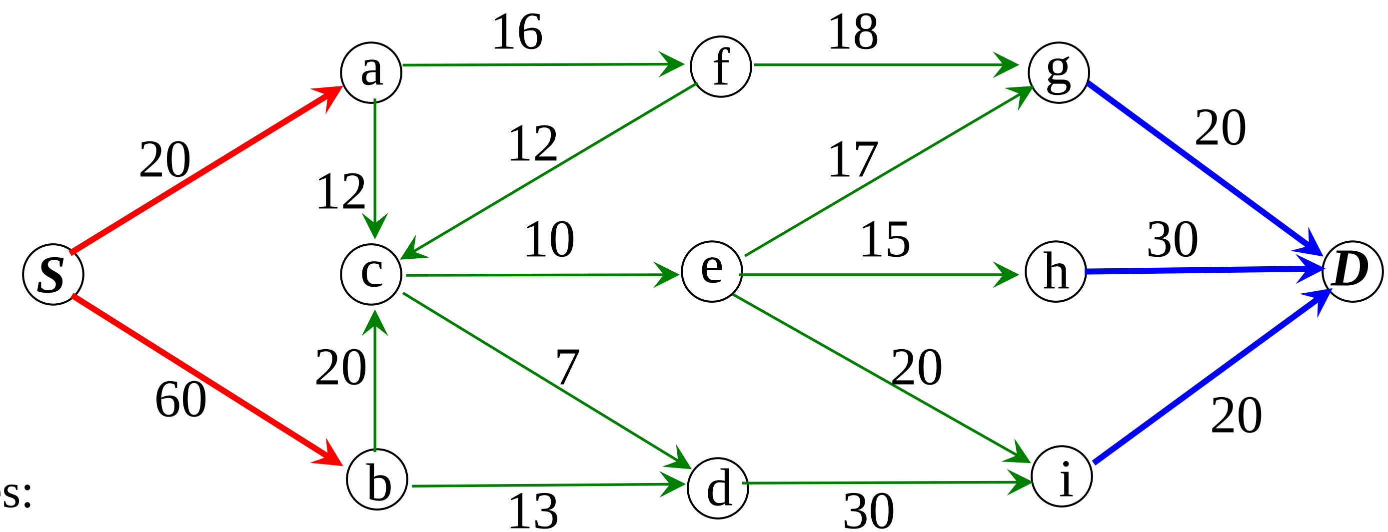
Graph Models and Applications

- Part I: Network and Flow
  - Transport networks and Maximum flow
  - Labeling algorithm
  - Matching in a bipartite graph
  - Existence Condition of Perfect Matching
- Part II: Graph Coloring
  - Vertex Coloring of Graph
  - Four-color and Five-color Theorem

A Model of Oil Supply

17 supplying capacity

17 cosuming capacity



Vertices:

a, *b*: refineries

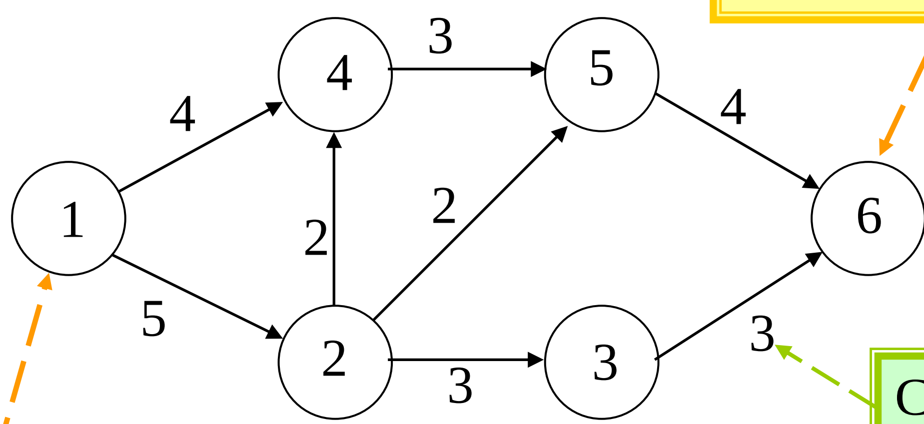
g, *h*, *i*: markets

others: relays

17 pipeline, with max capacity/week

Transport Networks：example

The unique node with out-degree 0
The **sink**



Capacity of edge, $C_{i,j}$

The unique node with in-degree 0
The **source**

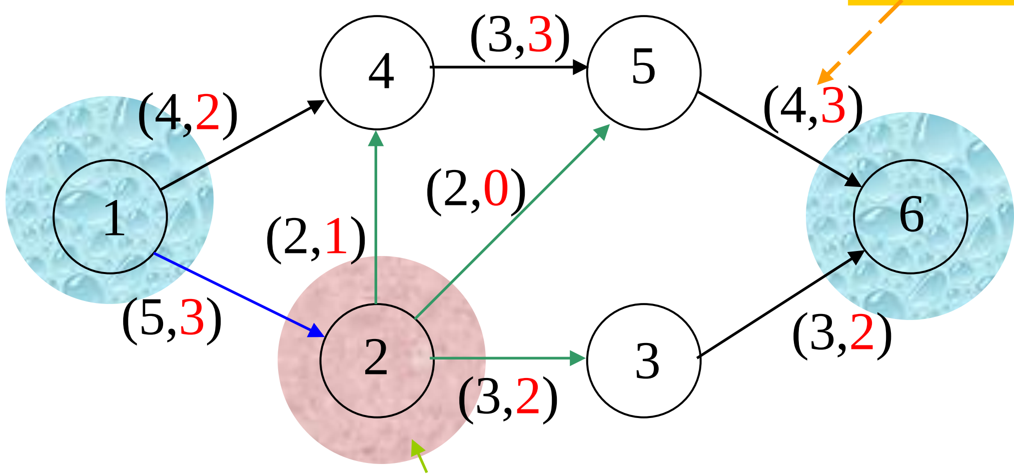
It is assume that all edges
are in one direction.

Transport Network

- A **transport network** is an ordered pair (G, k), where:
  - G is a weakly connected directed graph containing no loops
  - k is a nonnegative real function defined on E_G
  - There are two distinguished vertices S and D in V_G , and usually S has in-degree 0, called source, D has out-degree 0, called sink .

Flow：an example

($C_{i,j}$, $F_{i,j}$),
For any edge, $F_{i,j} \leq C_{i,j}$



Value of the flow is 5

Conservation (守恒) of flow:
Here: 3=1+0+2

Flow

- Let (G,k) be a transport network with source S and sink D. Assume the capacity function k is defined on the edges of G. A **flow** in G is a nonnegative real-valued function F defined on the edges of G such that:
  - [Capacity constraint] $0 \le F(e) \le k(e)$ for each edge $e \in E(G)$
  - [Conservation equation]

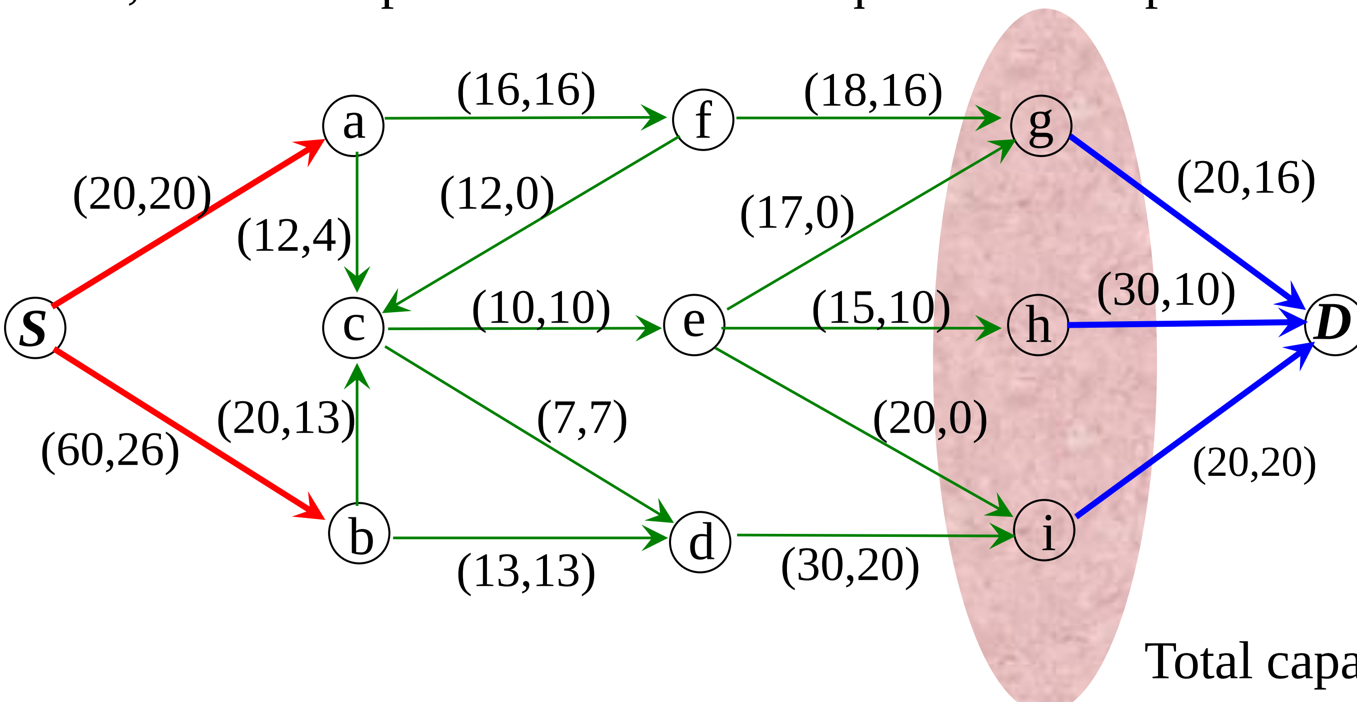
$$\sum_{y \in A(x)} F(xy) = \sum_{z \in B(x)} F(zx) \text{ for every } x \in V(G) - \{S, D\}$$

where $A(x) = \{y \mid xy \in E(G)\}$, $B(x) = \{z \mid zx \in E(G)\}$

Flow: One More Example

The actual transported amount on any road cannot exceed the edge capacity.

For any vertex, the total input amount must be equal to the output total.

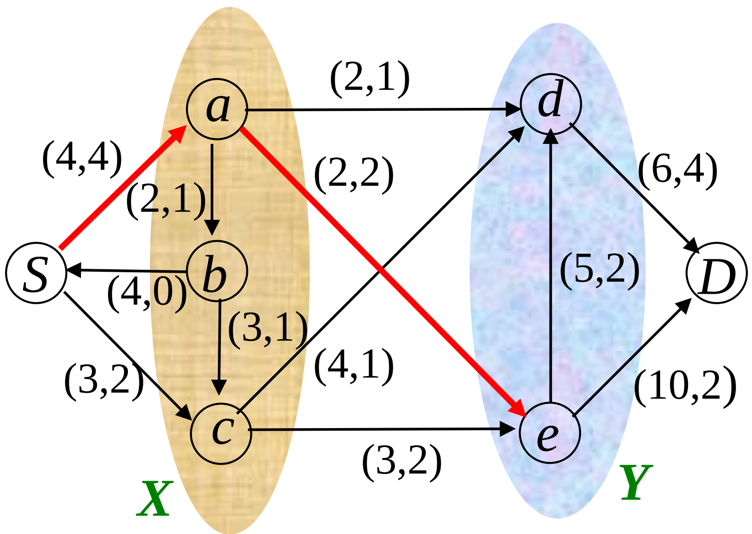


Total capacity: 70

Actual receipt: 46

The problem: Can we send more on the network?

Edge Sets in a Flow



$(X, Y) = \{ad, ae, cd, ce\}$

$k(X, Y) = 2+2+4+3 = 11$

$F(X, Y) = 1+2+1+2 = 6$

Note: $k(Y, X) = 0$, since $(Y, X) = \phi$

$(X, V_G) = \{bS, ad, ae, cd, ce\}$

$(V_G, X) = \{Sa, Sc\}$

Saturated edges

Unsaturated edges

Excess capacity(ce)=k(ce)-F(ce)=1

Value of Flow

- Let S and D be the source and sink, respectively, of a network (G,k). Let F be a flow. Then $F(S,V_G)=F(V_G,D)$

$$F(V_G,V_G)=\sum_{x\in V_G}F(x,V_G)=\sum_{x\in V_G}F(V_G,x)$$

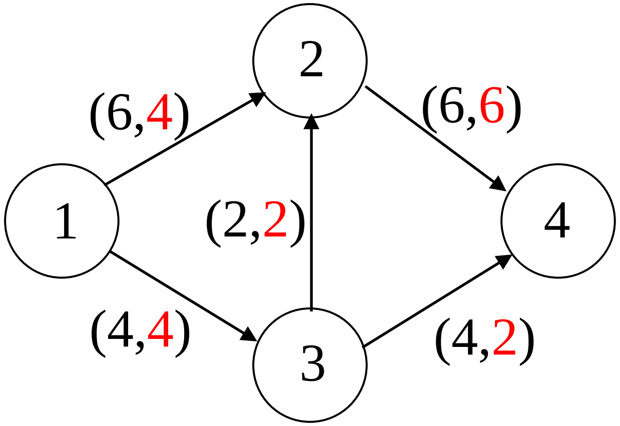
$$\text{However, } F(x,V_G)=F(V_G,x) \text{ for each } x\in V_G-\{S,D\}$$

$$\therefore \quad F(S,V_G)+F(D,V_G)=F(V_G,S)+F(V_G,D)$$

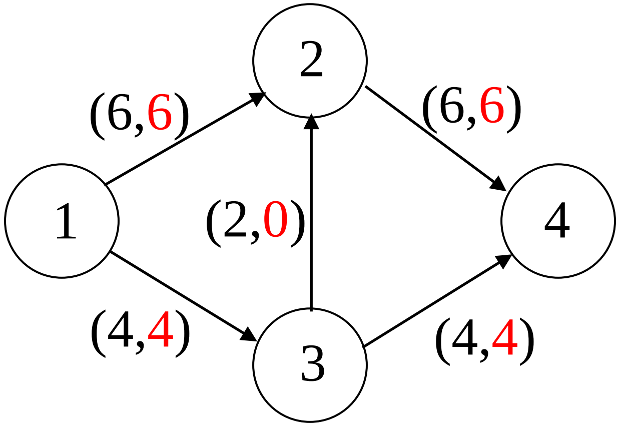
$$\text{Delete the terms equal to } 0:\ F(S,V_G)=F(V_G,D)$$

- The **value of a flow** F (denoted as $|F|$) is defined as the value of $F(S, V_G)$ (or, $F(V_G, D)$)

Flows with Different Value



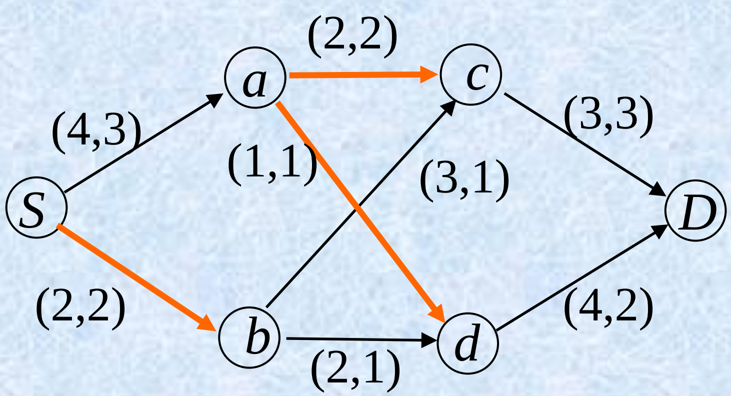
Value of flow: 8



Value of flow: 10

Maximum Flow

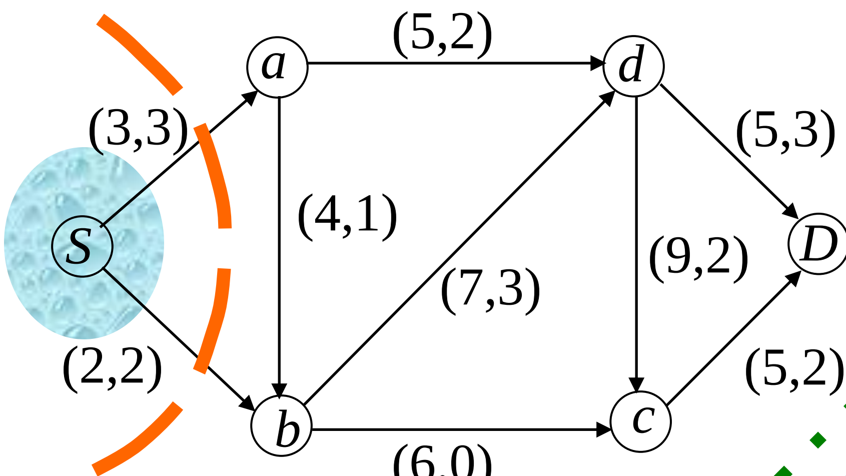
- A flow F in a network (G,k) is a **maximum flow** if $|F| \geq |F'|$ for every flow F' in (G,k)



A maximal flow

with value 5

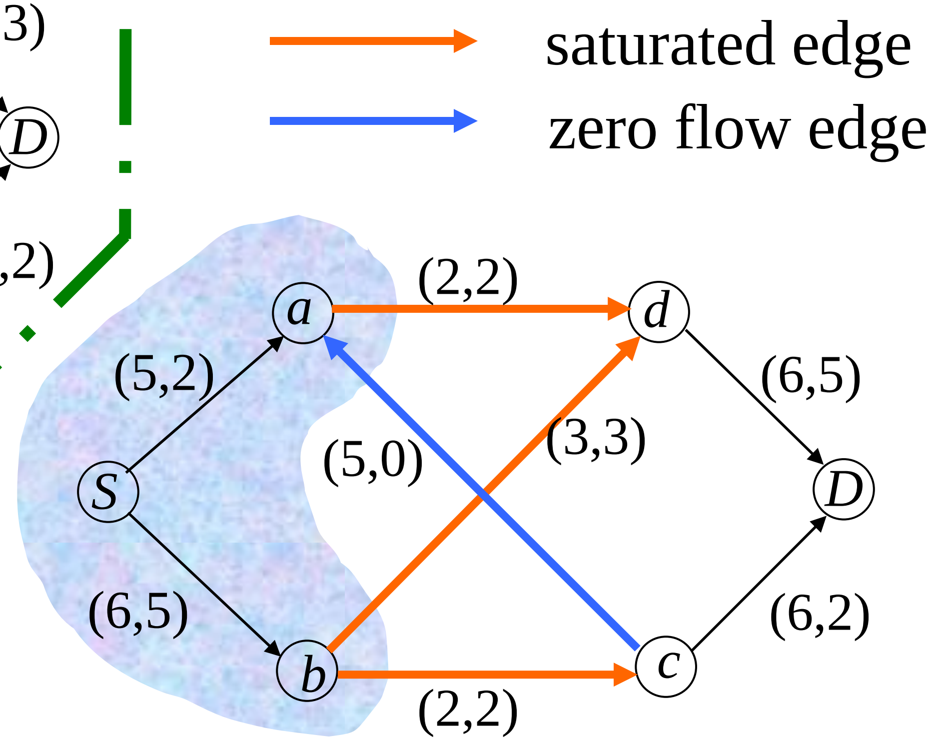
Two Examples of Maximum Flow



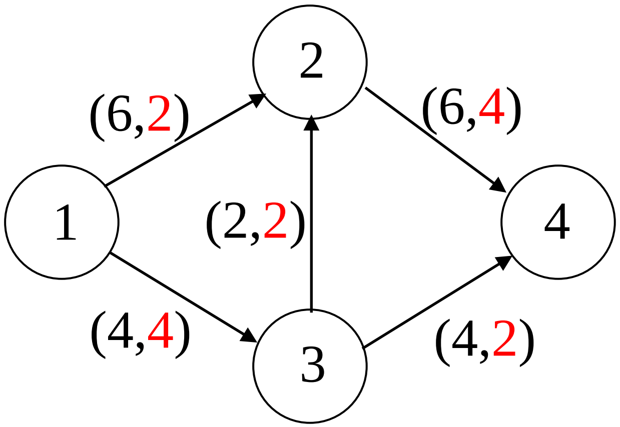
$X=\{S\}$

Minimal cut=$\{Sa,Sb\}$

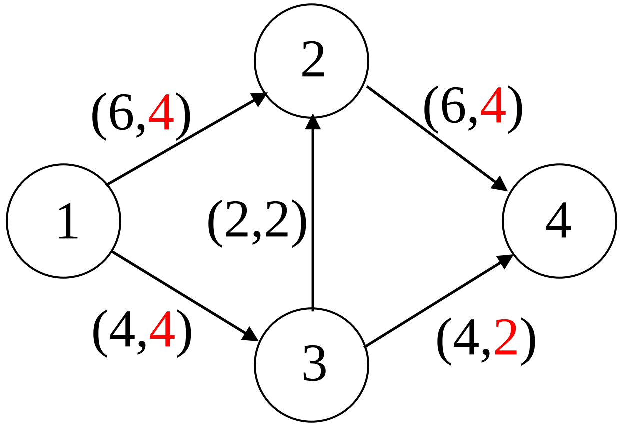
$X=\{S,a,b\}$



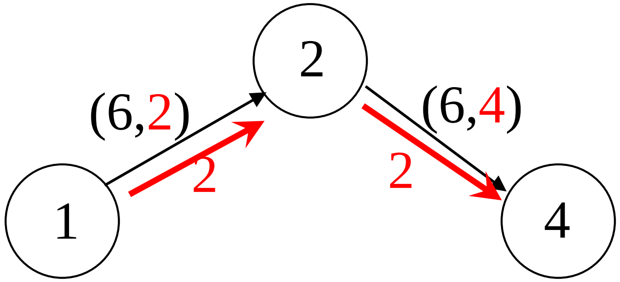
Problem of the Maximum Flows



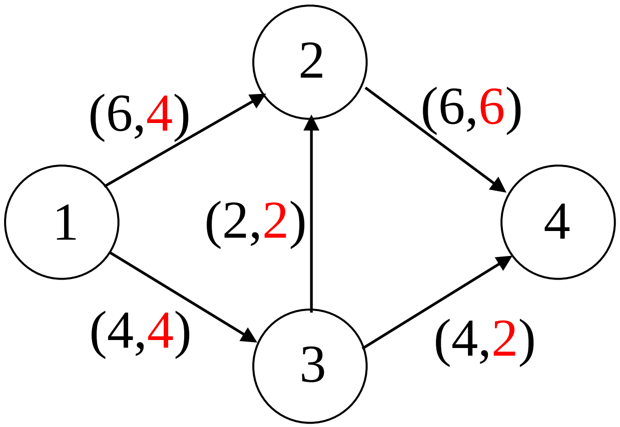
Value of flow: 6



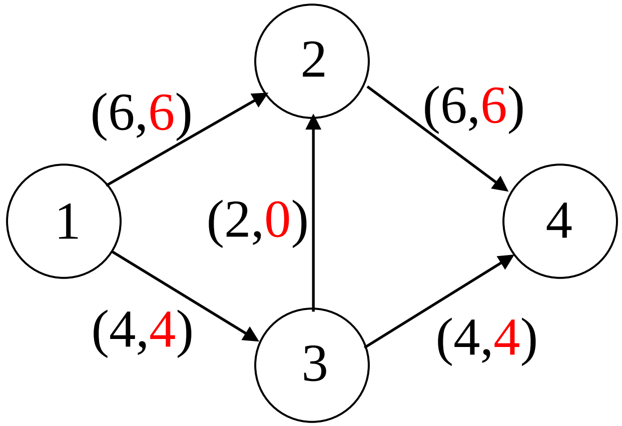
Value of flow: 8



Problem of the Maximum Flows

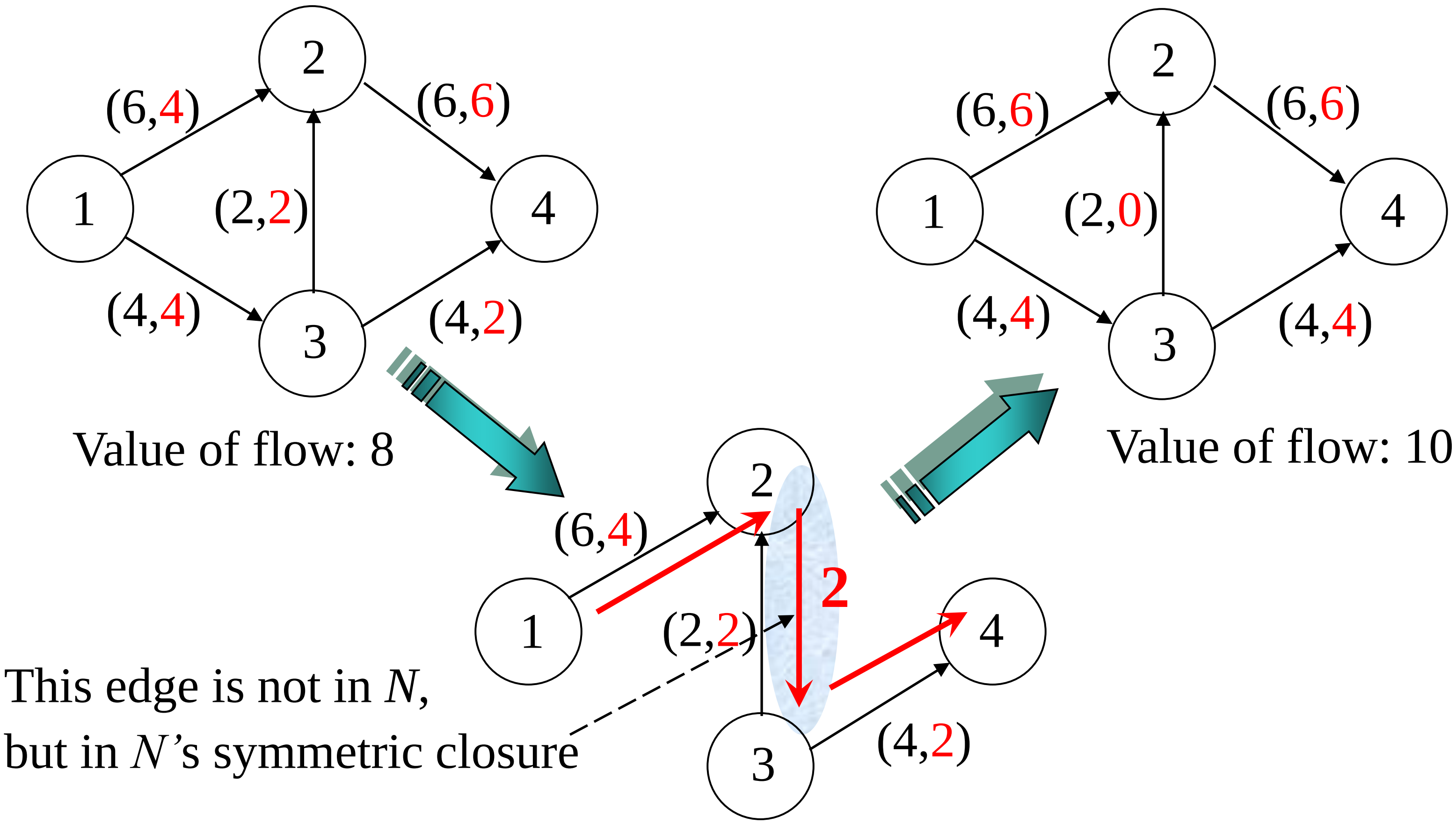


Value of flow: 8



Value of flow: 10

Maximum Flows

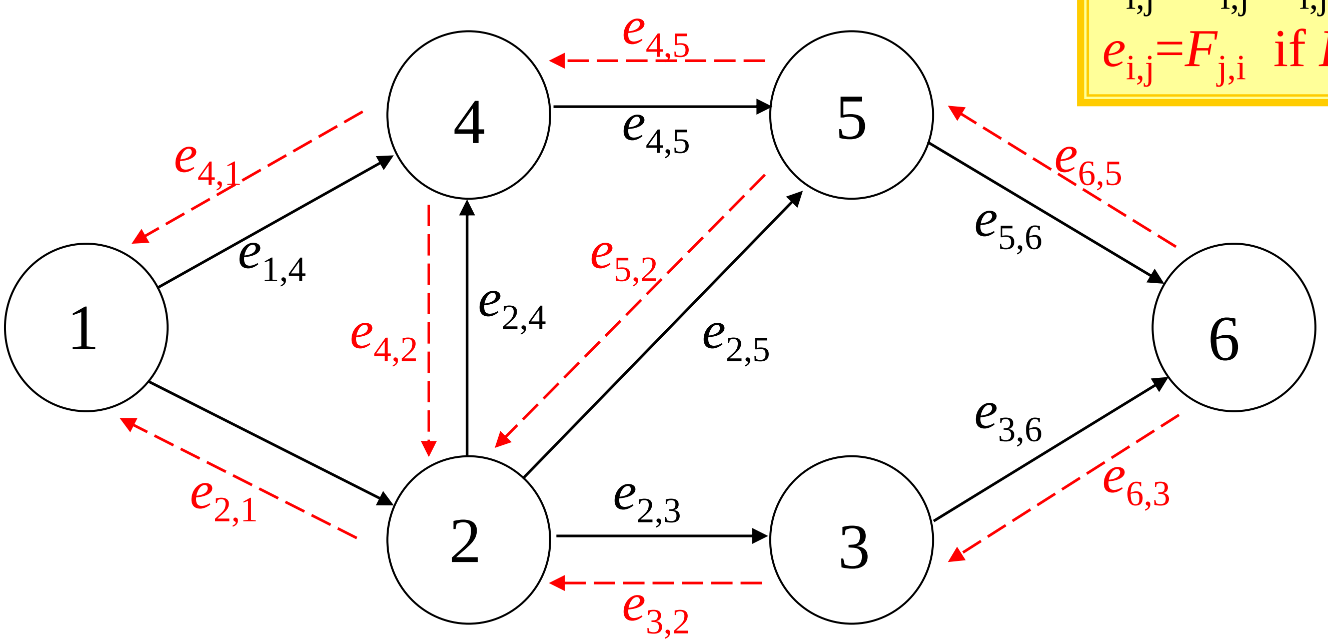


Genenal Senario

Excess 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)

Find a path from Source to Sink such that all edges with a positive excess capacity.

1. The value of flow can be increased by the minimal excess capacity of the edges.
2. Conservation constraints
3. Capacity constraints

Labeling Algorithm (Ford & Fulkson)

Initialization: set all flow to 0

Step 1: (1) Identify N_1

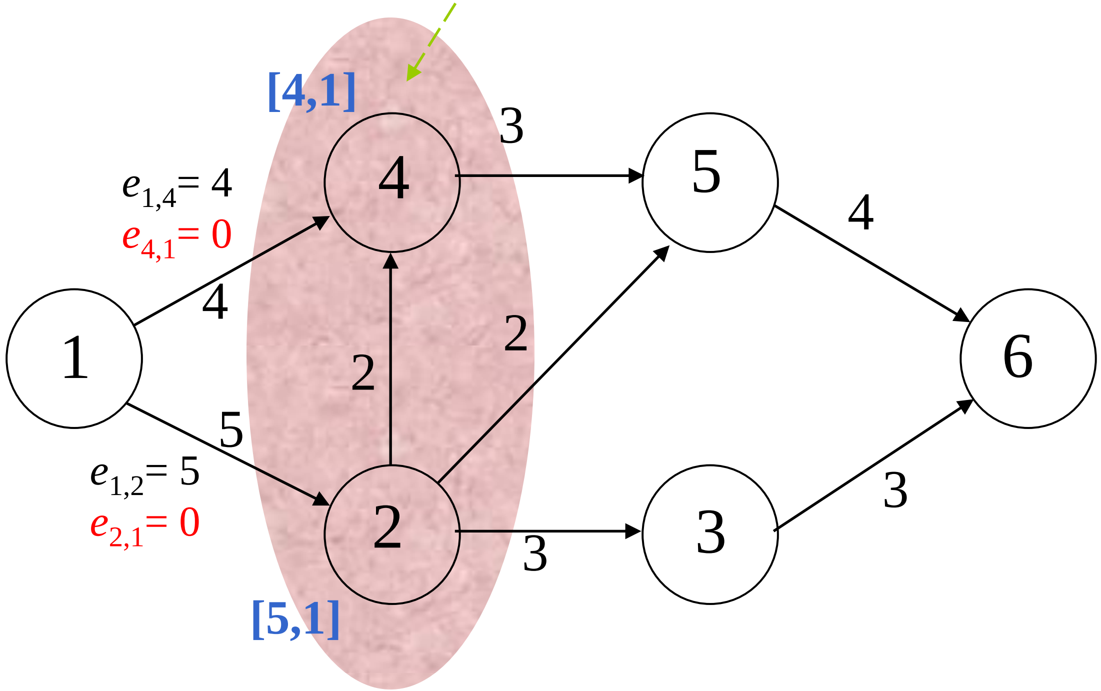
(2) Label nodes in N_1 as follows

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

Excess capacity of $(1,j)$

$[E_j, 1]$

Denoting node 1



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 $N_2(j)$ as follows

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

$[E_k, j]$

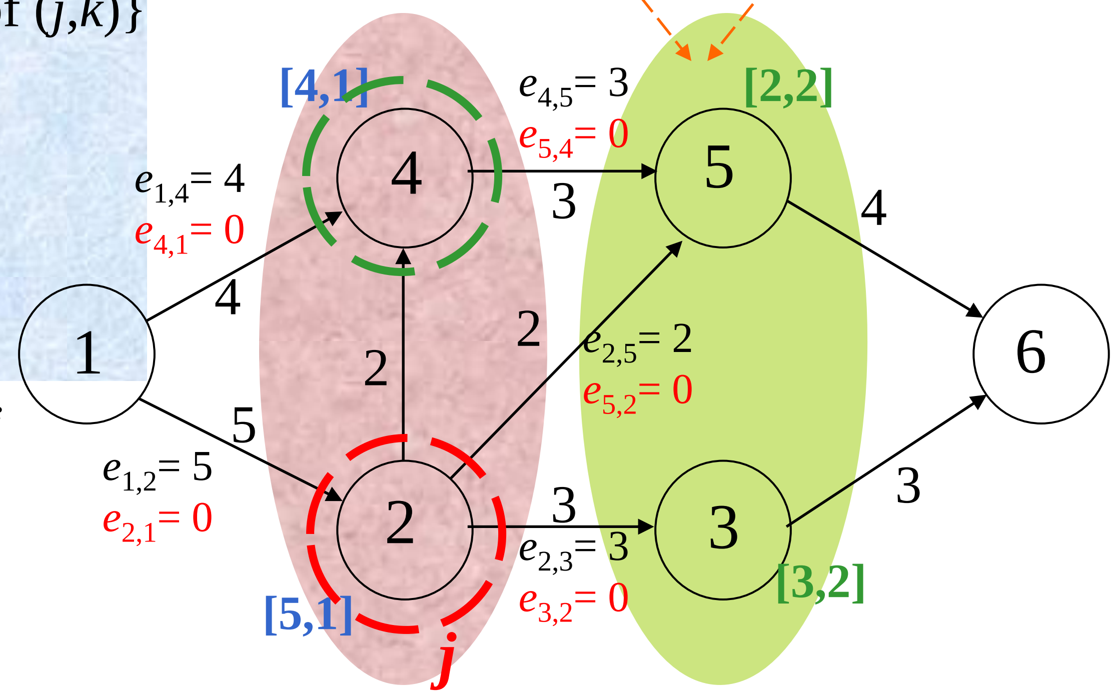
Denoting node j

(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 excess capacity

Also N_2, in this case



Labeling Algorithm (Ford & Fulkson)

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

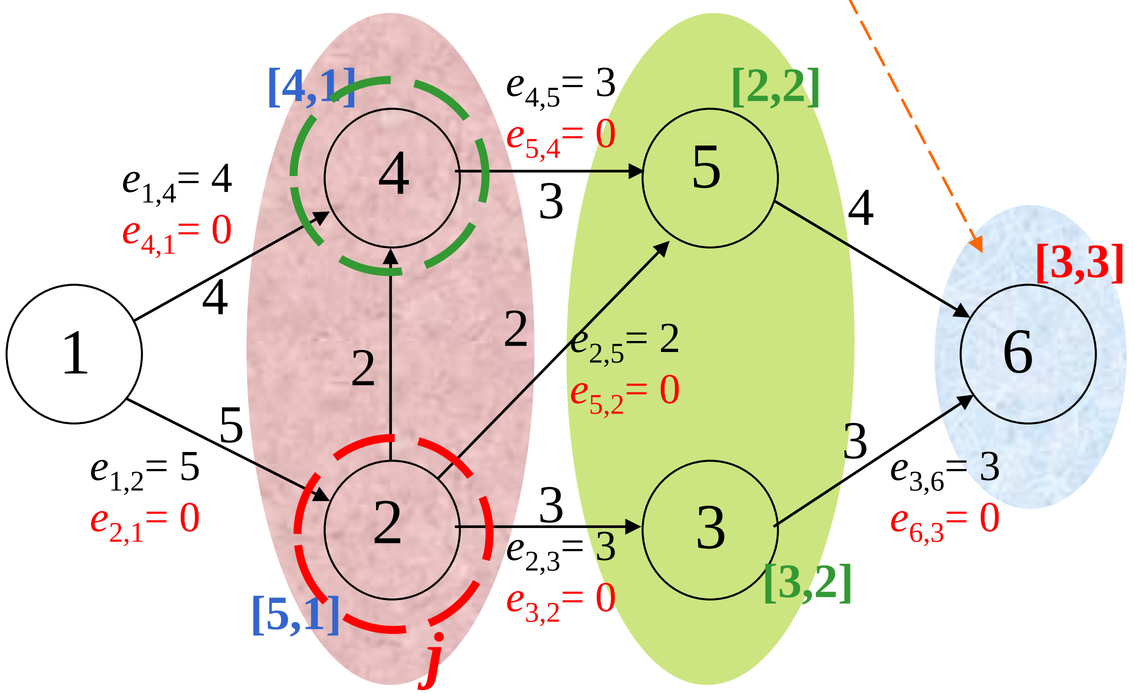
(i) the sink has been labeled, goto step 4

or

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

(note: the source is not labeled)

N_3, the sink has been labeled



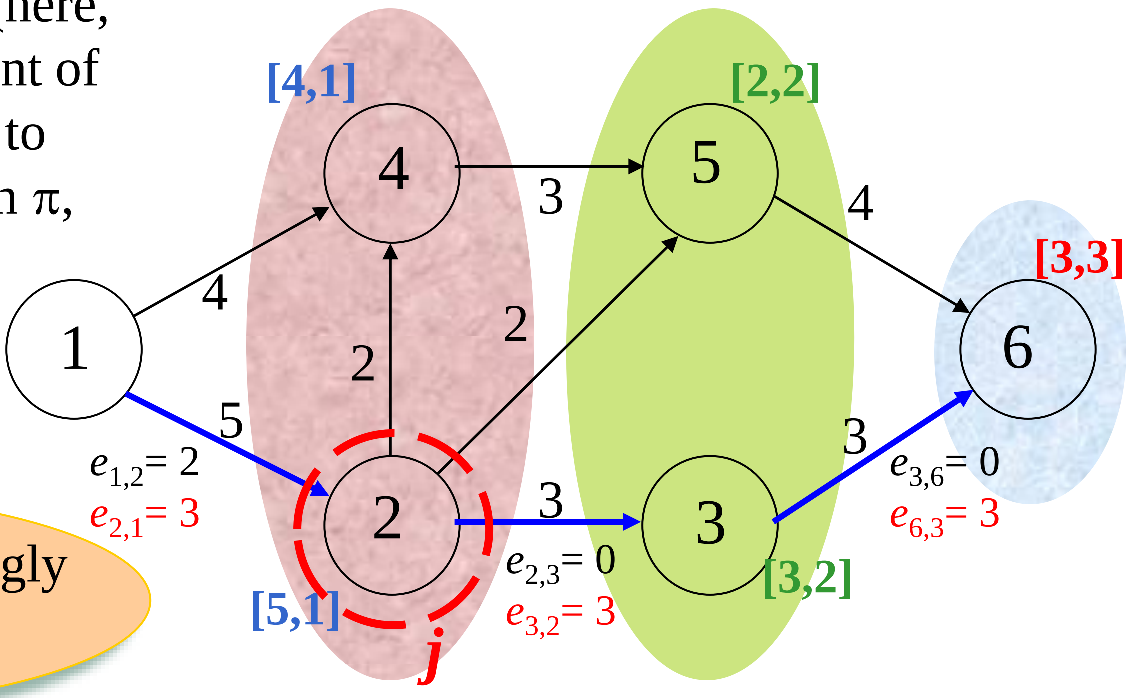
Labeling Algorithm (Ford & Fulkson)

Situation after one full cycle:

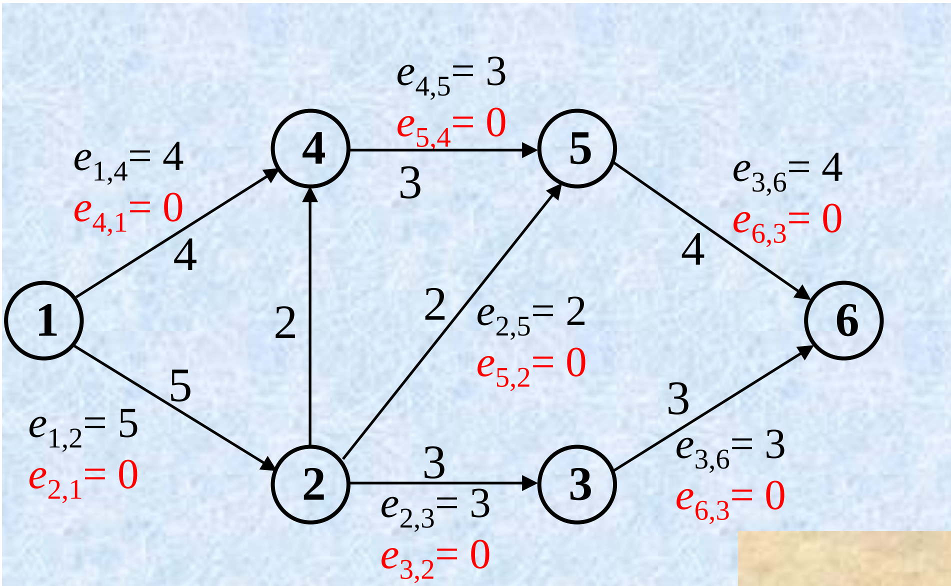
The label of sink is [E_n, m] (here, [3,3]), where E_n is the amount of extra flow that can be made to reach the sink through a path π, and the path can be traced backward by node m

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

and then return to step 1

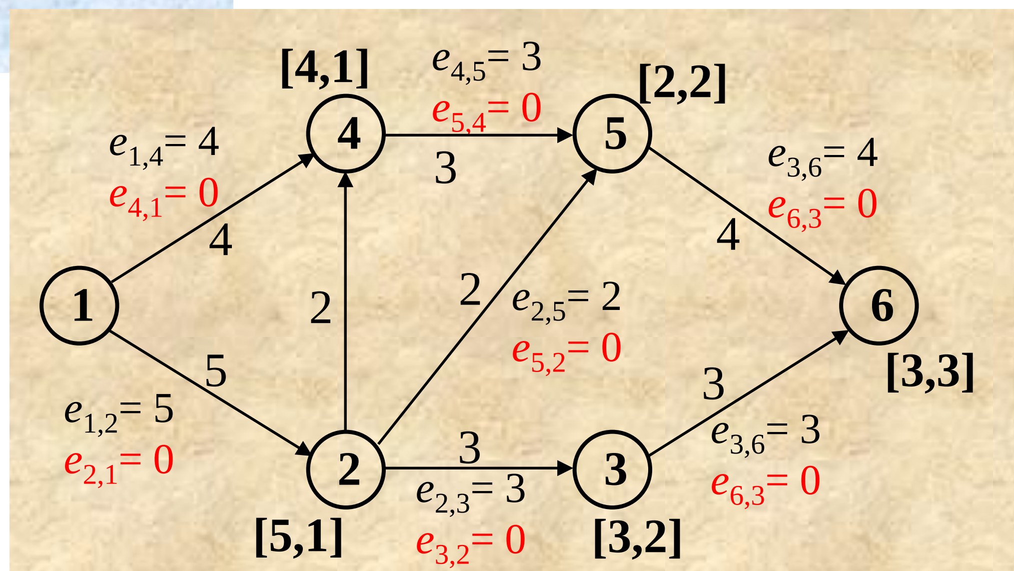


Applying Labeling Algorithm

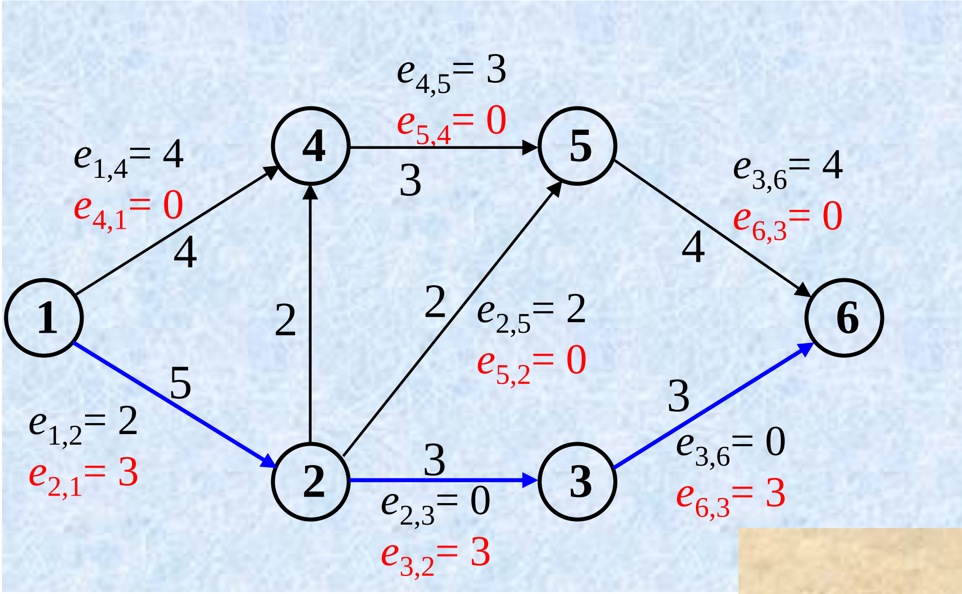


At the beginning, setting all flow to 0

After the first cycle

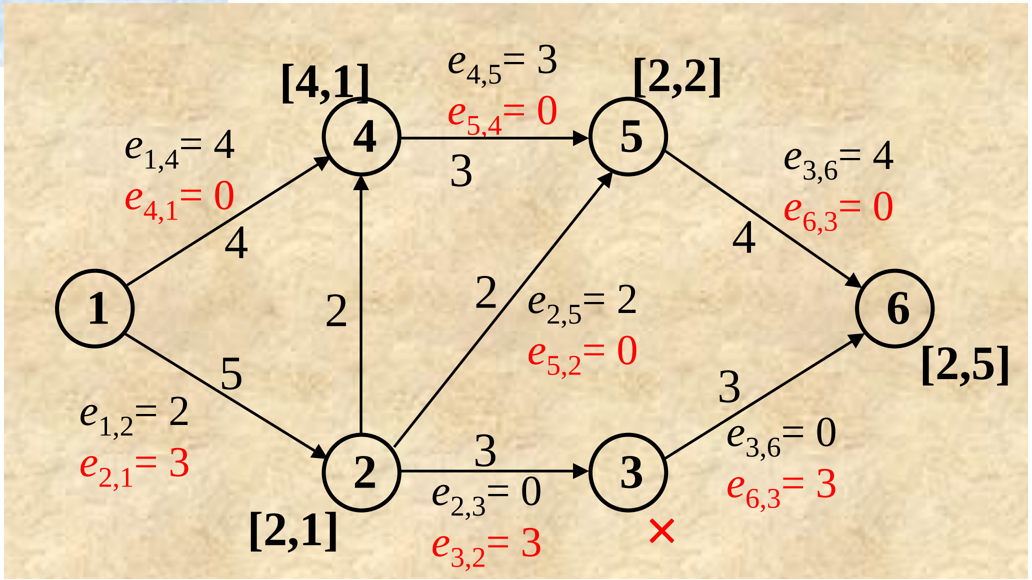


Applying Labeling Algorithm

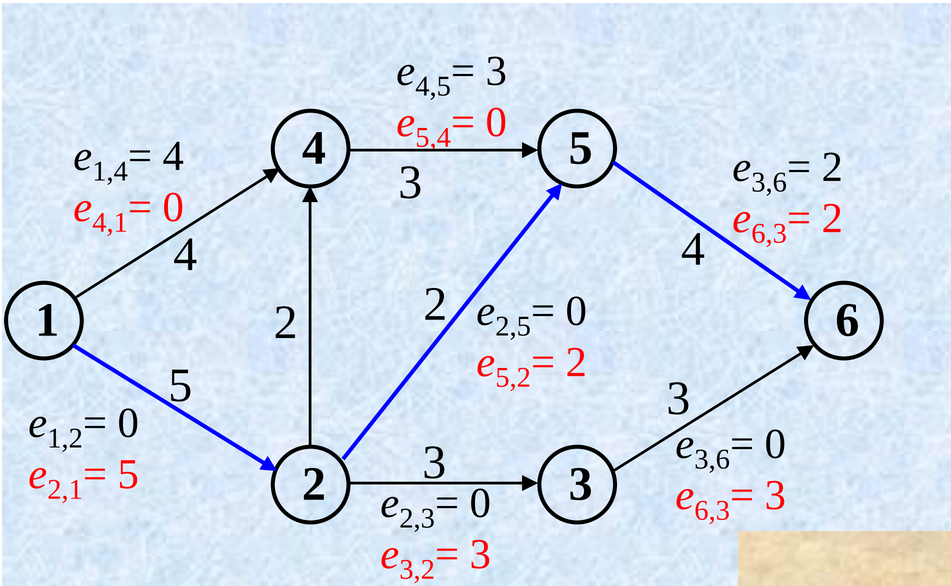


After the first cycle

After the second cycle

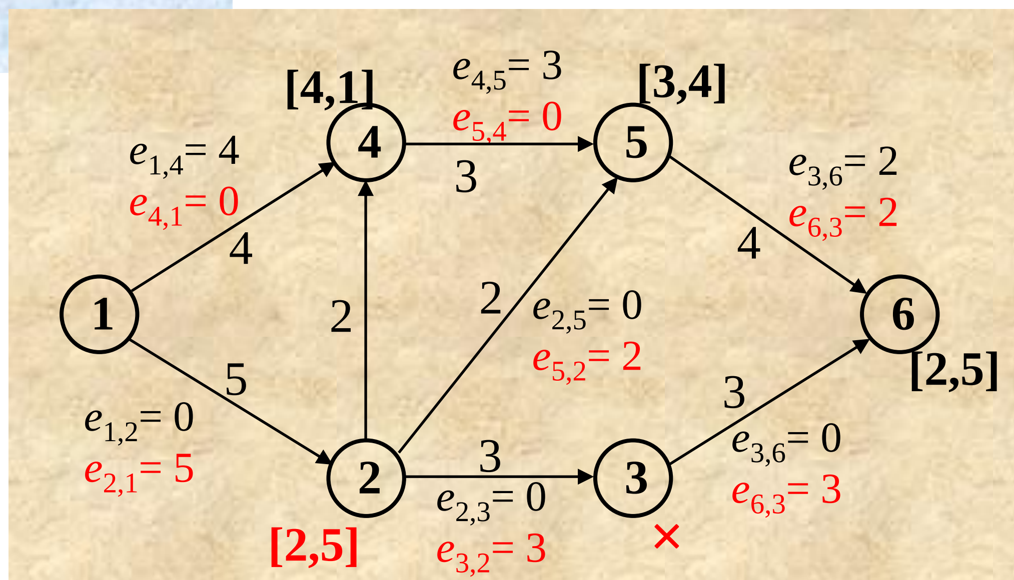


Applying Labeling Algorithm

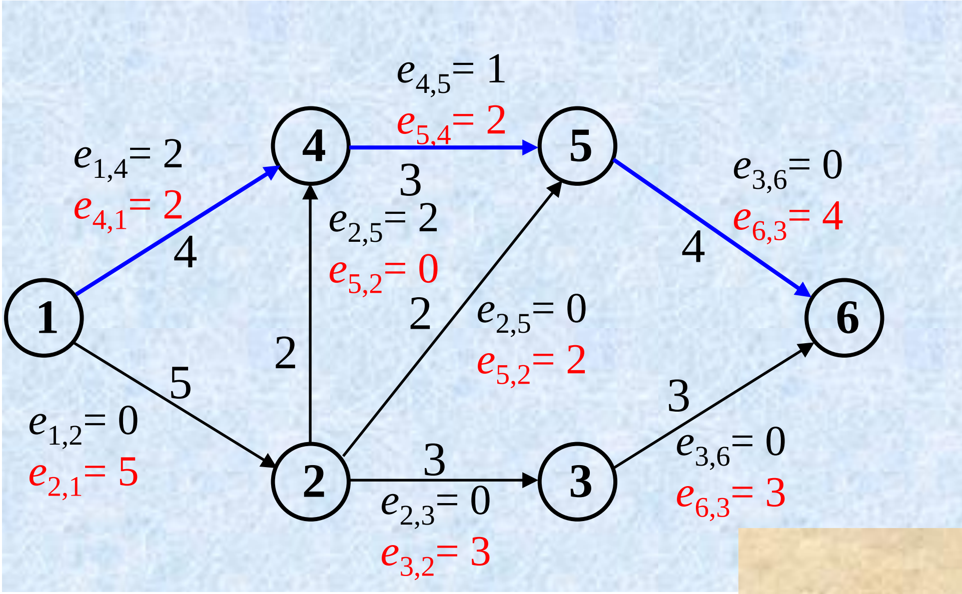


After the second cycle

After the third cycle



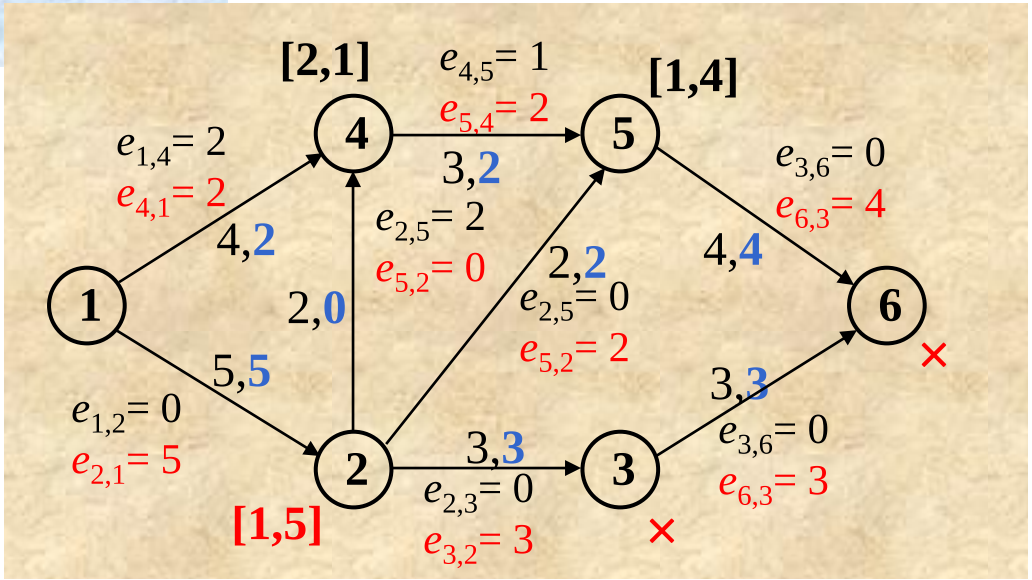
Applying Labeling Algorithm



After the third cycle

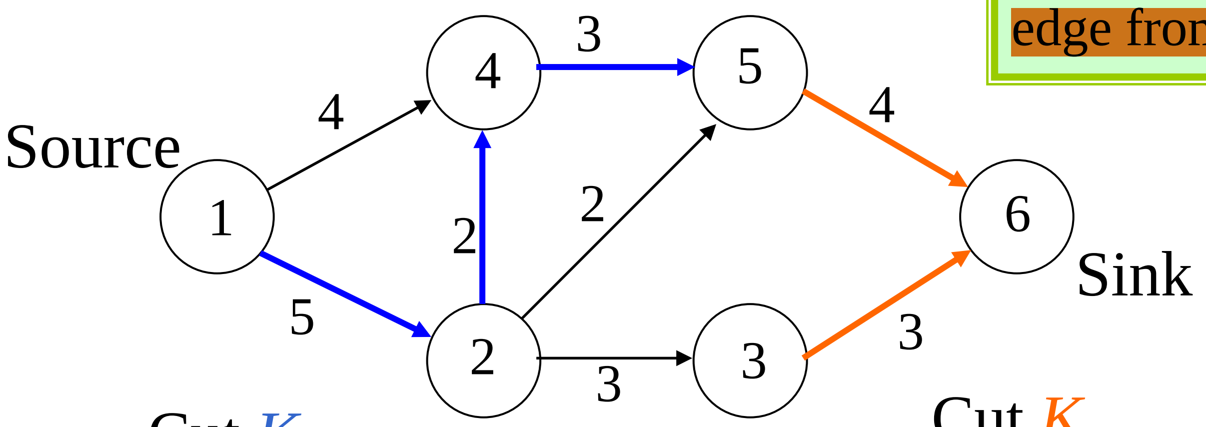
After the fourth cycle

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



Flow and Cut

Cut: a set K of edges in a network N, having the property that **every** path from the source to the sink contains **at least one** edge from K.



Cut K_1

$c(K_1)=8$

Cut K_2

$c(K_2)=7$

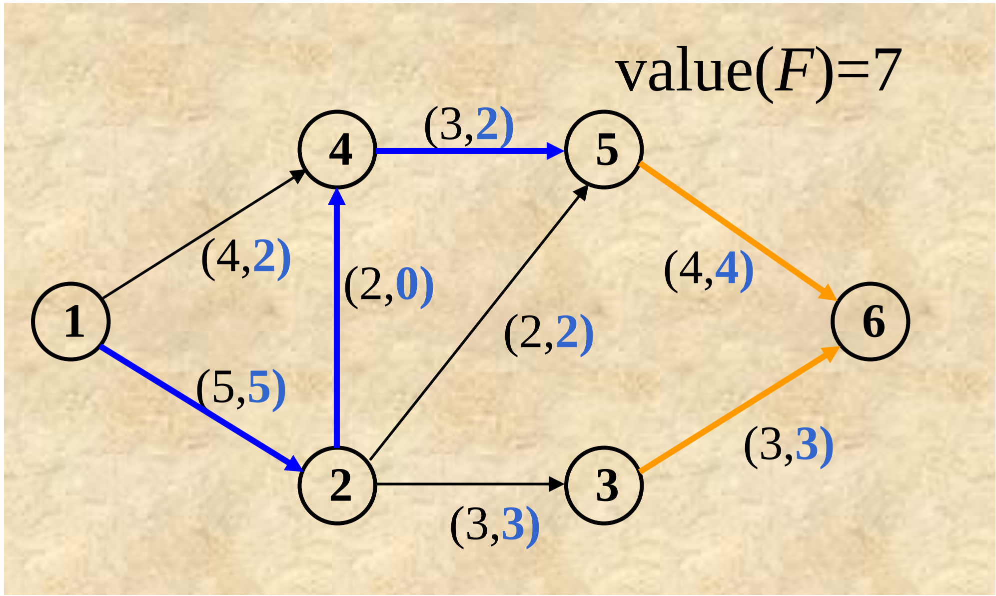
Max Flow Min Cut Theorem

For any flow F, and any cut K, all parts of F must pass through the edges of K. Since $c(K)$ is the maximum amount that can pass through the edges of K, so, value(F)$\leq c(K)$.

If value(F)=$c(K)$, then the flow uses the full capacity of all edges in K, F must be a flow with maximum value, and, on the other hand, K must be a cut with minimum capacity.

Theorem

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



Correctness of Labeling Algorithm

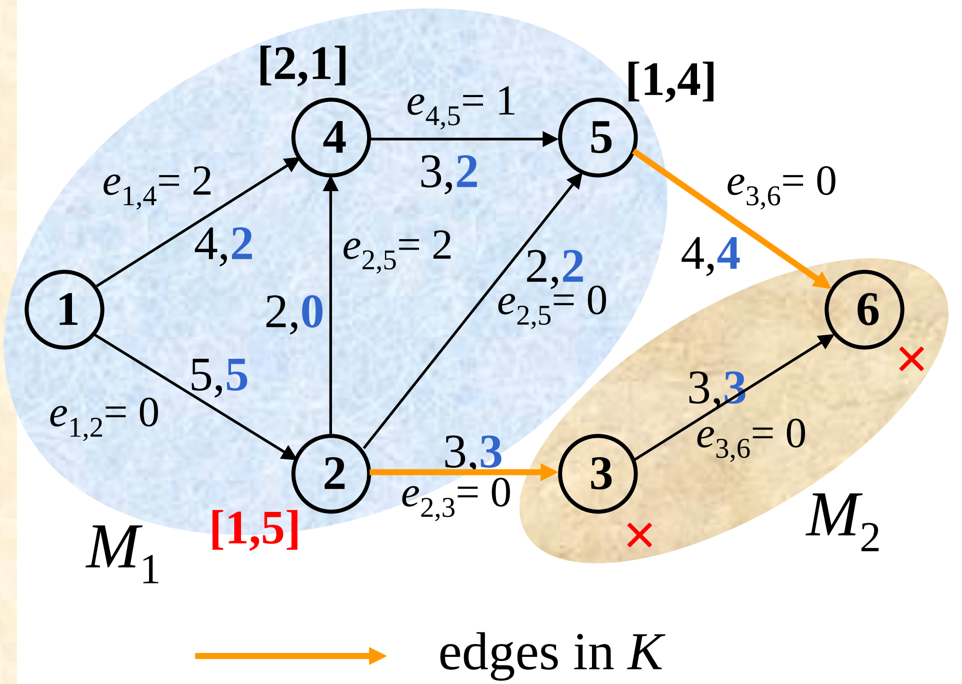
M_1 : labelled nodes
M_2 : other nodes
K : all edges from M_1 to M_2.

Any path π from source to sink contains an edge begins with node in M_1 and ends with node in M_2. So, K is a cut.

For (i,j) in K, (i,j) carrying its full capacity in the final flow F.
In the final flow:
1、no flow from M_2 to M_1
2、for nodes in M1 other than source, the conservation rules applied.

So, $value(F)$ = c(K)

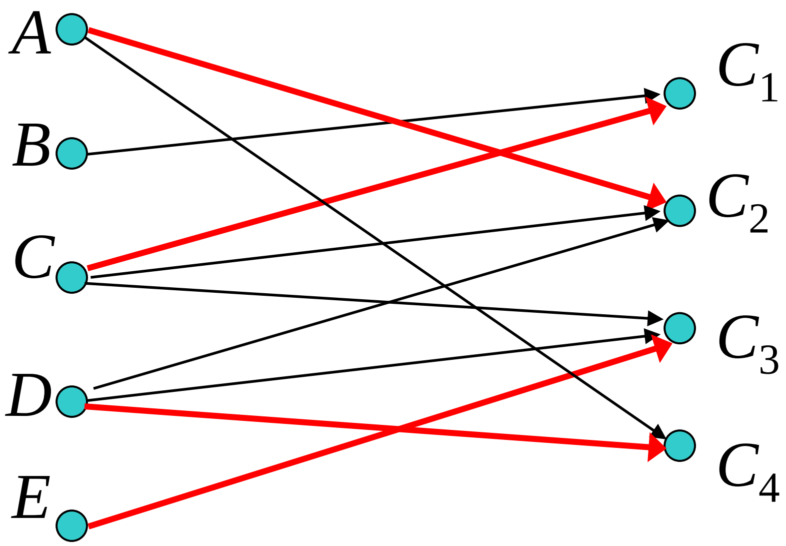
Algorithm stops at Step 4



"Non-Transport" Transport Network

5 persons A, B, C, D, E belongs to 4 committees C_1, C_2, C_3, C_4, where $C_1=\{B, C\}$; $C_2=\{A, C, D\}$; $C_3=\{C, D, E\}$; $C_4=\{A, D\}$.

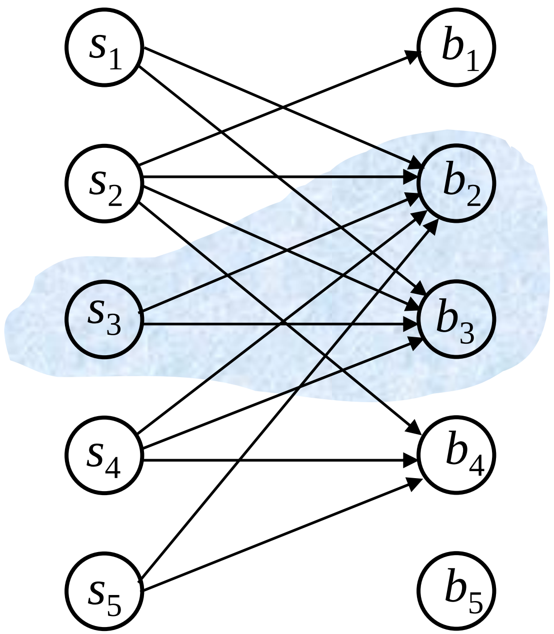
Is it possible to select 4 chairperson where no one chairs more than one?



Assignment Problem

- The assignment problem
  - A organization has n positions to fill and m applicants. Each applicant has a list of qualifications which make him suitable for certain positions.
  - Is it possible to assign each applicant to a position to which he or she is suitable?
  - If not, what is the largest number of people that can be assigned to the positions?
  - How should these assignments be made?

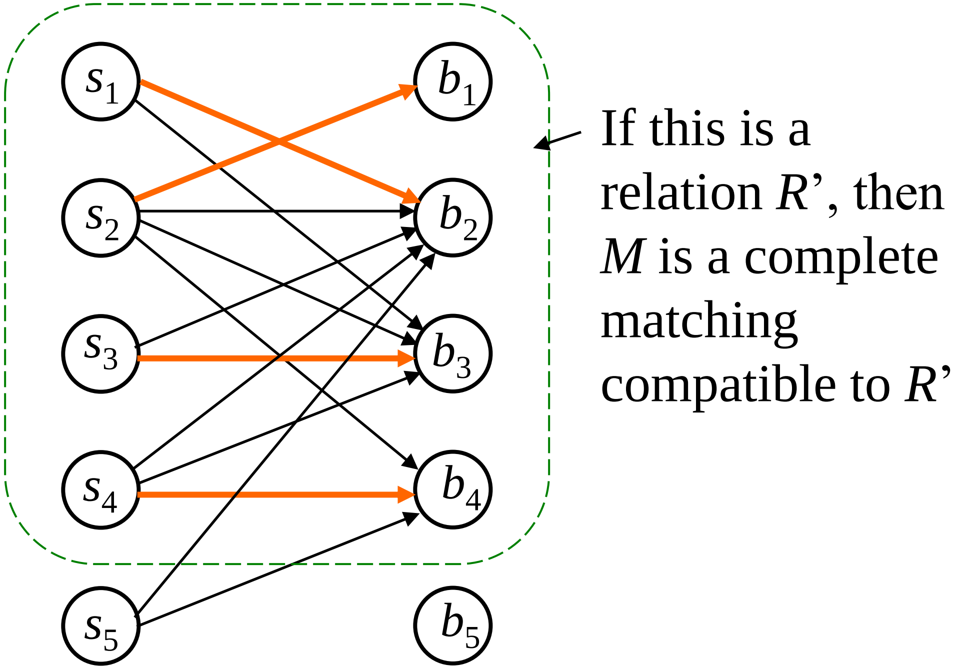
Matching



Relation R

Note: $R(s_3, b_2)$, and $R(s_3, b_3)$,

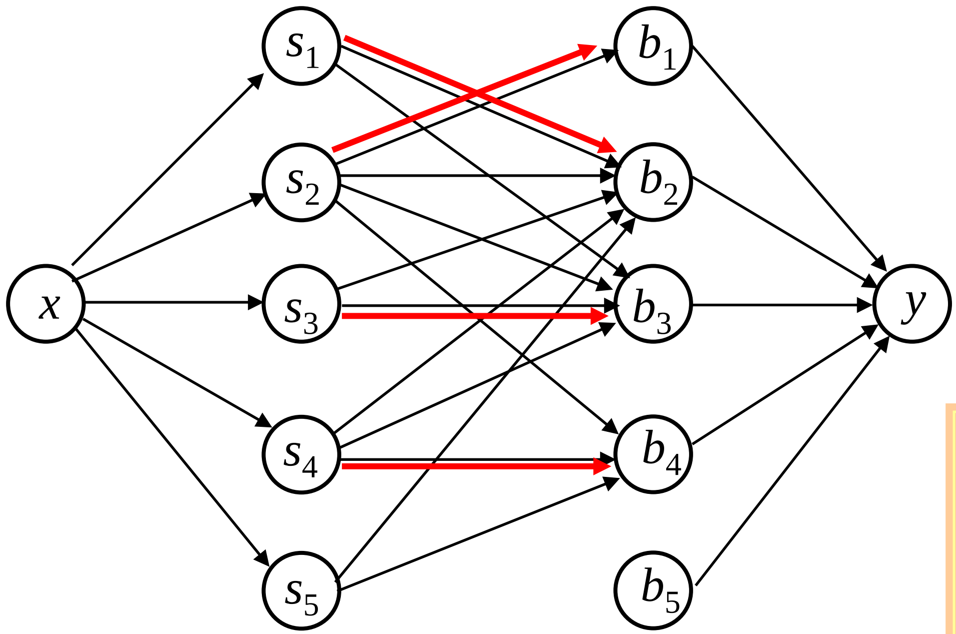
$R(\{s_3\})=\{\ b_2, b_3\}$



Matching function M, compatible to R

A maximal matching, but not complete

Matching and Flow in Network



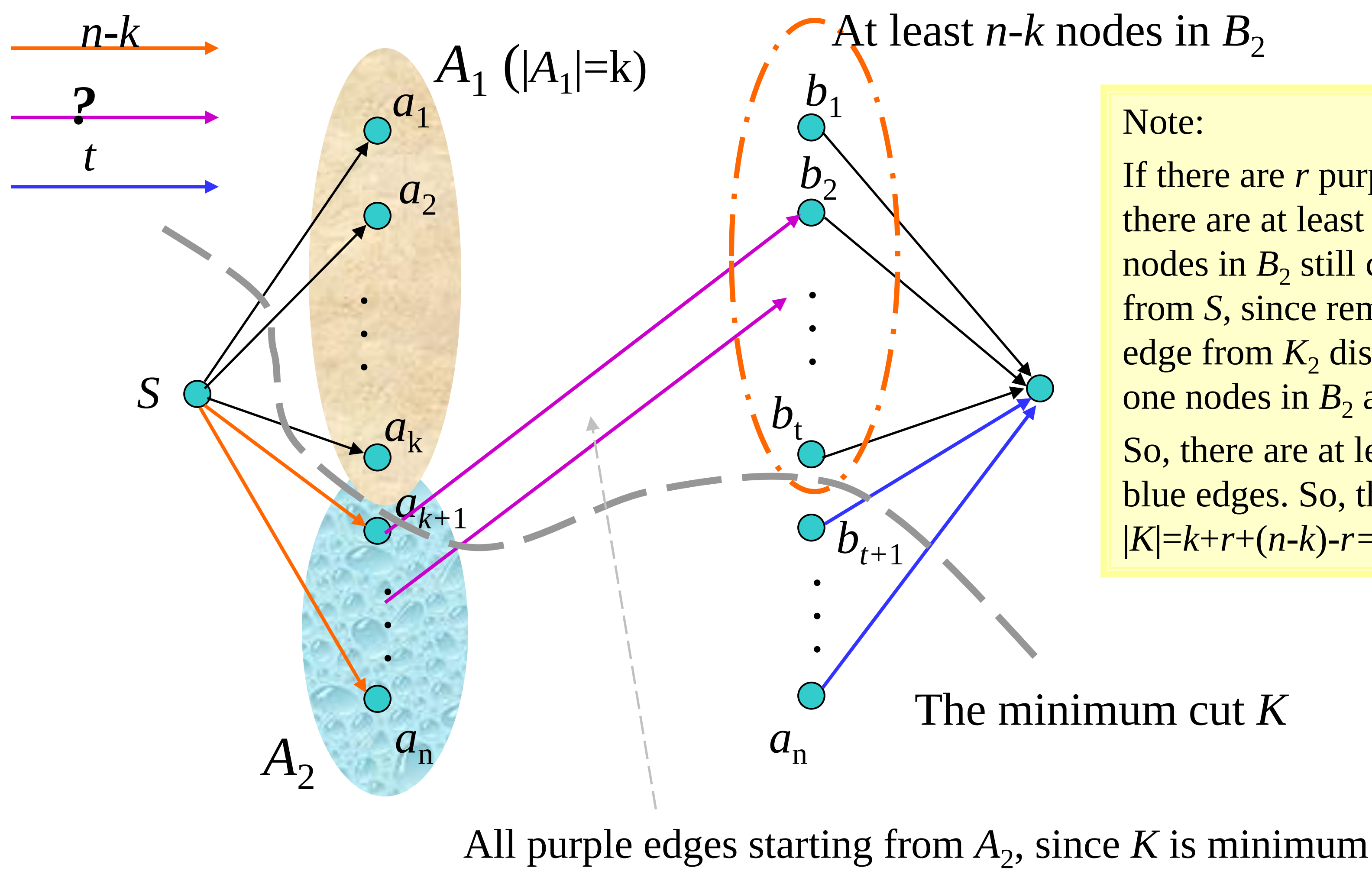
with each capacity set to 1

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 $X \subseteq A$, $|X| \leq |R(X)|$
- Proof:
  - $\Rightarrow$ Obviously
  - $\Leftarrow$ Show that the minimum cut in N has value $n=|A|$.
    - Suppose K is a minimal cut.
    - Consider all edges in K as in three sets
      - S_1: those begin at supersource;
      - S_2: those correspond to pairs in R;
      - S_3: those end at supersink.
    - Considering the situation of removing the three sets one by one, we can see that K contains at least n edges.

Proof of Hall's Theorem:



Note:

If there are r purple edges, there are at least (n-k)-r nodes in B_2 still connected from S, since removing one edge from K_2 disconnecting one nodes in B_2 at most.

So, there are at least (n-k)-r blue edges. So, the $|K|=k+r+(n-k)-r=$n

Chromatic Number of Graph

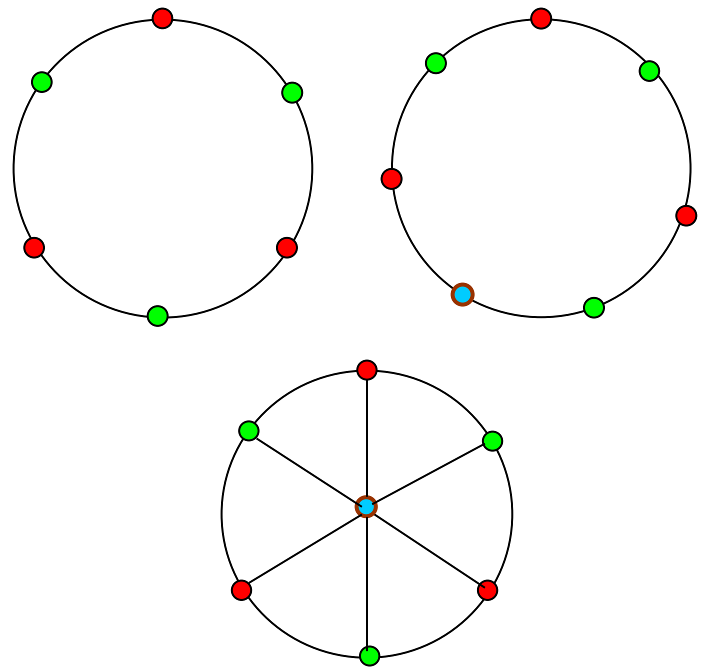
- Definitions
  - Coloring each vertex in a graph without rings, if no adjacent vertices has the same color, the coloring is called a proper coloring.
  - The smallest number of colors needed to produce a proper coloring of a graph G is called the chromatic number of the graph, denoted by $\chi(G)$.

“Commonsense” about $\chi(G)$

- $\chi(G) \leq |V_G|$, and only when $G=K_n$ the equality holds
- If H is a subgraph of G， If $\chi(H)=k$, then $\chi(G) \geq k$。
- If $d(v)=k$, then all the vertices adjacent to v can be properly colored in at most k colors.
- The chromatic number of G is the chromatic number of the largest component of G.

Chromatic Number: Examples

- $\chi(G)=1$ iff. there is no edges in the graph
- $\chi(K_n)=n$
- If G is a cycle:
  - If $|V_G|=2k$, $\chi(G)=2$
  - If $|V_G|=2k+1$, $\chi(G)=3$
- If G is a wheel
  - If $|V_G|=2k$, $\chi(G)=4$
  - If $|V_G|=2k+1$, $\chi(G)=3$



Chromatic Number of Bipartite Graph

- G is a bipartite graph if there is a two-set partition of the set of vertices such that all edges in the graph connect members from different sets in the partition.
- G is a bipartite graph if and only if there is no odd cycle in G.
- G is a graph with at least one edge (i.e. not discrete graph), then, $\chi(G)=2$ if and only if G is a bipartite graph
  - $\Rightarrow$ if $\chi(G)=2$， then there is no odd cycle in G;
  - $\Leftarrow$ if G is a bipartite graph, then only one color is needed for one set in the partition.

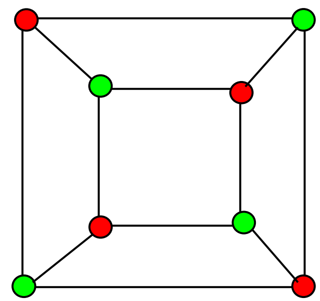
Chromatic Number: Examples

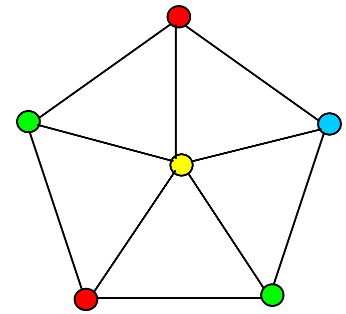
- Upper left:

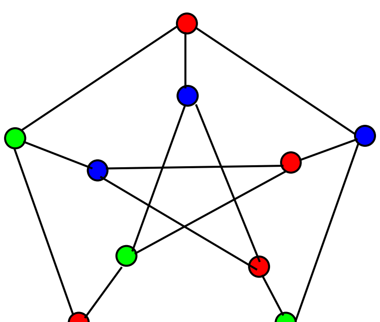
  a bipartite graph
- Upper right: wheel
- Lower left

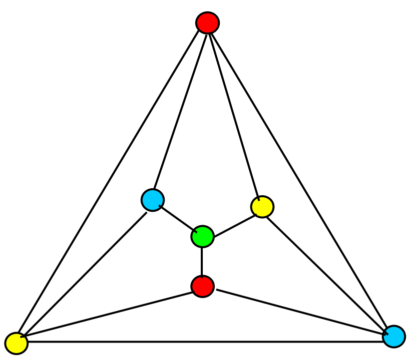
  $\Delta=3$， $\therefore \chi=3$
- Lower right

  $\Delta=4$， but $\chi>3$



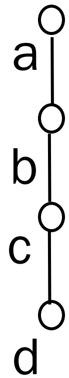


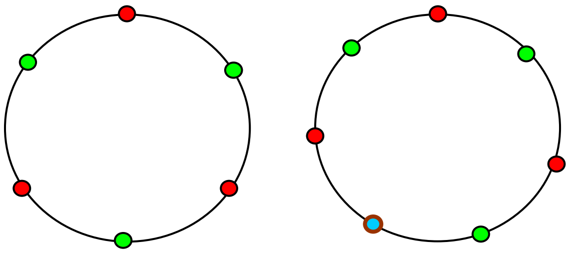


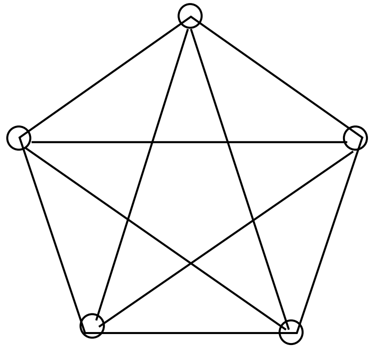


Chromatic Polynomial

- Give a graph G and a set of "color" ($\{c_1,c_2,\ldots,c_n\}$), the number of different coloring for G is a function of n.
- The function is a polynomial function, called the chromatic polynomial







$P_G(x)=x(x-1)^3$

$P_G(x)=(x-1)^n + (-1)^n(x-1)$

$P_G(x)=x(x-1)(x-2)(x-3)(x-4)$

Recursive Formula

$$P_G(x) = P_{G_e}(x) - P_{G^e}(x)$$

where $G_e = G - \{e\}$, and,

G^e is the graph by merging the edge e in G.

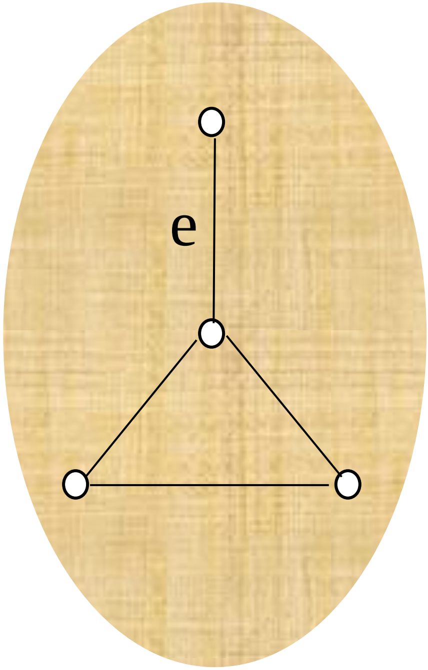
Proof:

Compare the coloring of G and G_e.

For G_e, any coloring in which the two endpoints of e are the same color is not in the coloring of G.

Howerver, the number of such coloring scheme is just that of G^e.

An Example



G^e is K_3, so, the polynomial is:

$$x(x-1)(x-2)$$

G_e has two component, one a single vertex, and the other, K_3. So, the polynomial is:

$$x(x(x-1)(x-2))=x^2(x-1)(x-2)$$

$P_G(x) = x^2(x-1)(x-2) - x(x-1)(x-2) = x(x-1)^2(x-2)$

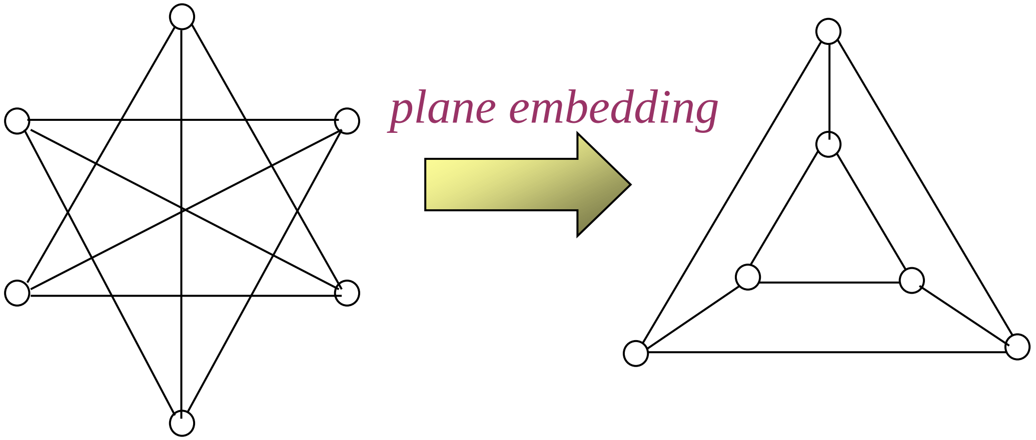
$\chi(G)=3$

Concept of Planar Graph

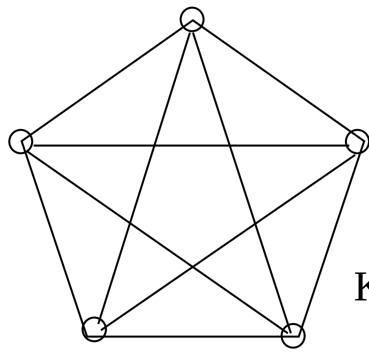
- Embedding a graph on a plane: Drawing a digram for a graph, such that no two edges cross unless on the endpoint.
- A graph is a planar graph if it has plane embedding

Notes：

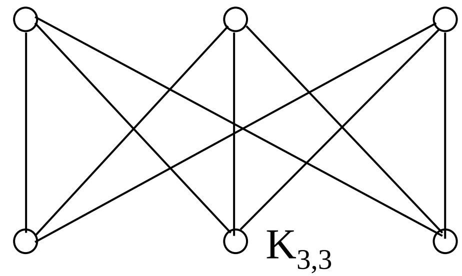
- Acyclic graph is planar.
- A graph is non-planar, if any of its subgraph is.
- Unconnected graph can be considered by components.



Typical Non-planar Graph

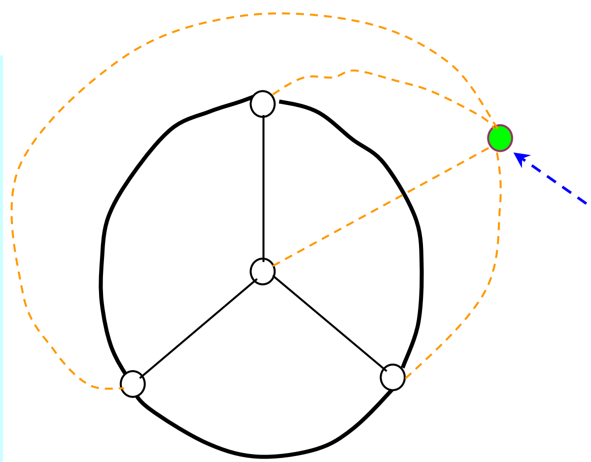


K_5



$K_{3,3}$

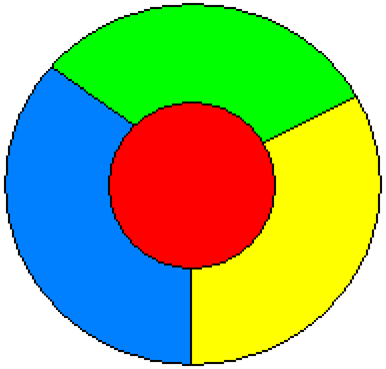
Jordan theorem: a closed curve C divides the plane into 2 parts: the inner and the outer. The line connecting two vertices located in the two sections respectively must intersection with C.



Jordan condition ocuurs, wherever the vertex is placed

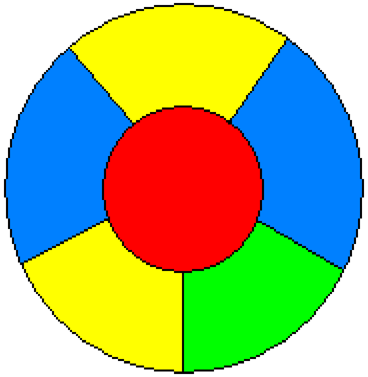
Francis Guthrie's Conjecture

Coloring areas in a map, such that no two areas with common border have the same color, four colors is enough.



3 colors can't do.

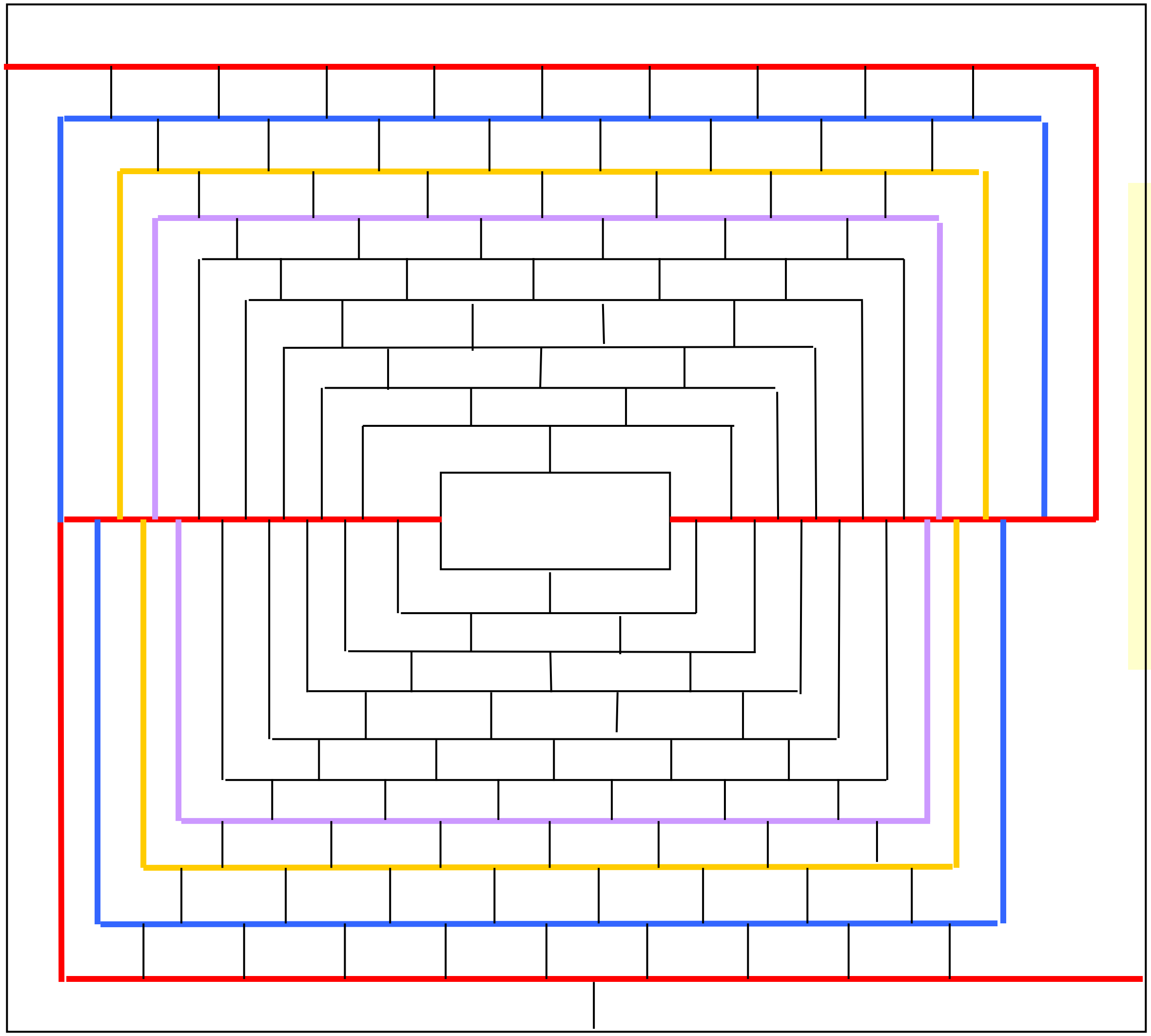
- Guthrie, de Morgan



It can be proved that no such pattern exits: 5 sections, with each adjacent to all other 4.

But …

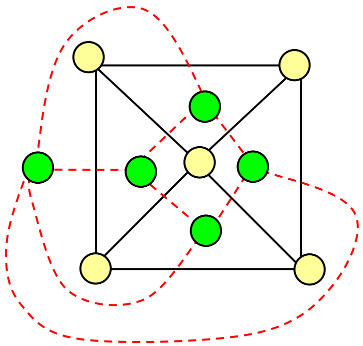
Martin Gardner's Gift for Fool's Day

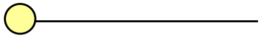


A counterexample for four-color conjecture ?

Scientific American
Fool's Day of 1975

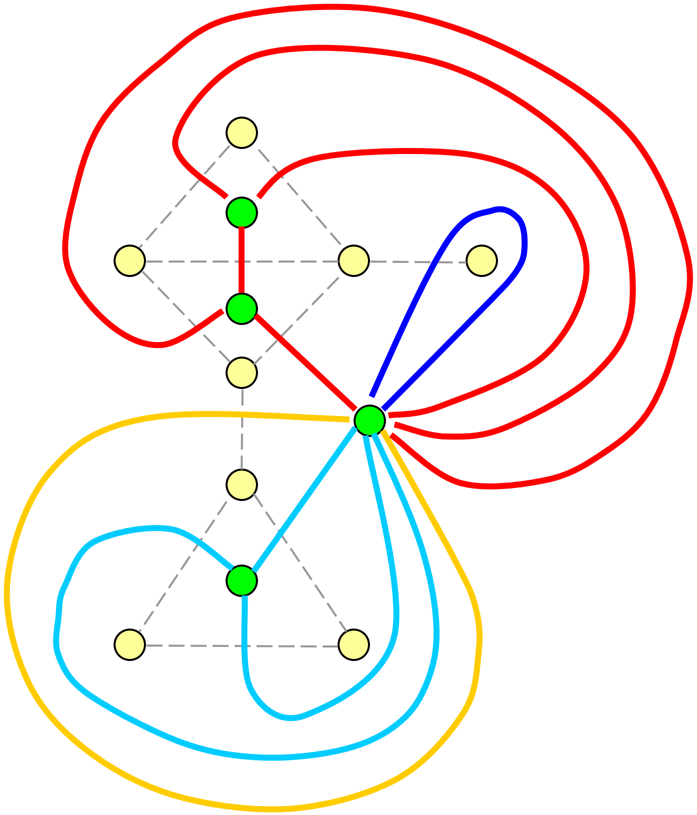
Dual Graph of a Planar Graph





Vertices and edges in G

Vertices and edges in the dual graph of G



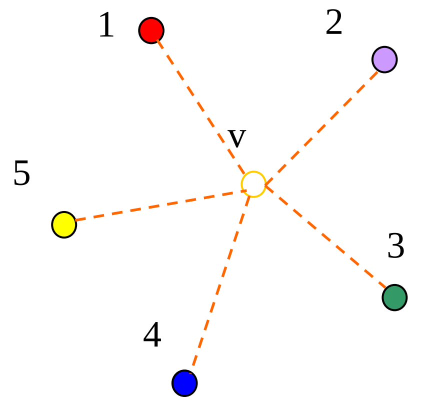
Five Color Theorem

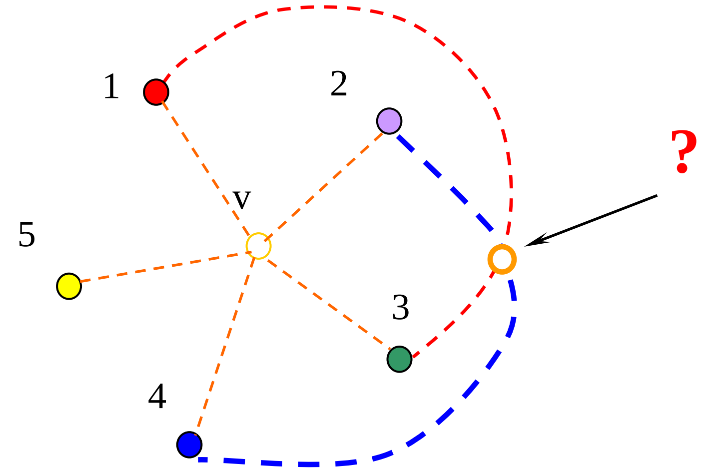
- Coloring a planar map, with different colors for adjacent areas, five colors is enough.
- Proof (sketch)

  There must be a 5-degree vertex in any simple planar.

  Induction on the number of vertices (if n≤5, obviously)

  induction as follows:





Home Assignments

- To be checked
  - pp.328: 5-11, 14, 19-21
  - pp.333: 4, 5, 8, 10, 14-19
  - pp.338: 15, 16, 19, 23, 26, 27
- Self tests