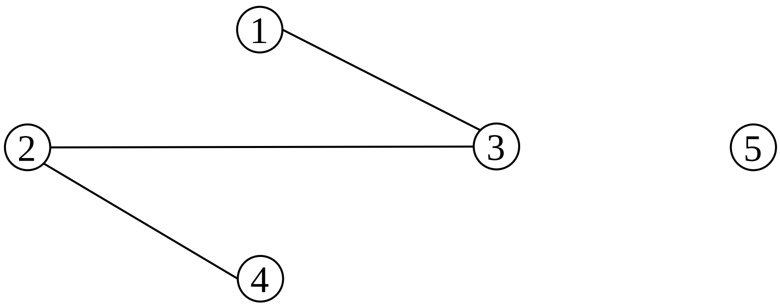
Preliminaries

- **Graphs**

***G* = (*V*, *E*),** ***V* : set of vertices**

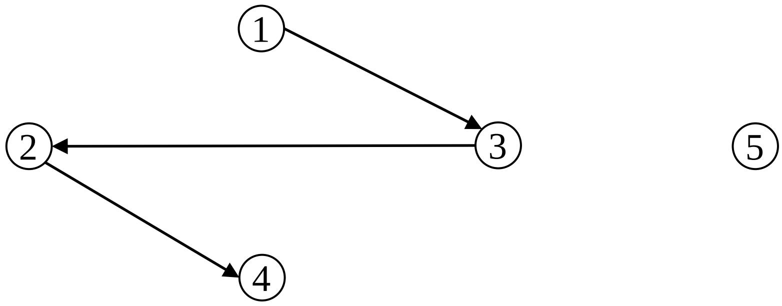
***E* : set of edges (arcs)**

(Undirected) Graph : (*i*, *j*) = (*j*, *i*) (edges)



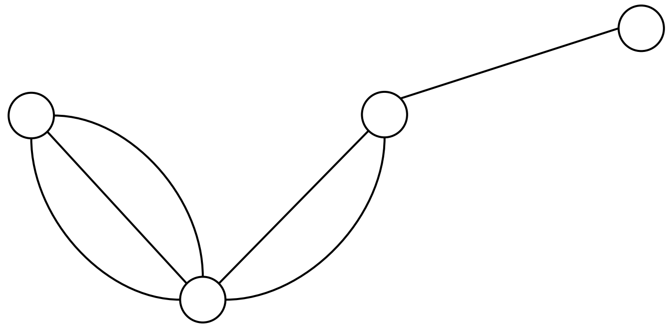
$$V = \{1, 2, 3, 4, 5\}, \quad E = \{(1, 3), (3, 2), (2, 4)\}$$

Directed Graph (Digraph) : $<i, j> \neq <j, i>$ **(arcs)**



$V = \{1, 2, 3, 4, 5\}, \quad E = \{<1, 3>, <3, 2>, <2, 4>\}$

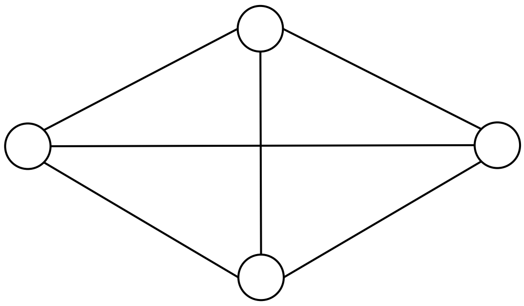
Multigraph : ***E*** **is a multiset of edges**



Loop : (i, i) **or** $\langle i, i \rangle$

Simple Graph : **no loops and no two edges connecting the same pair of vertices**

Complete Graph : **each pair of distinct vertices is connected by an edge**

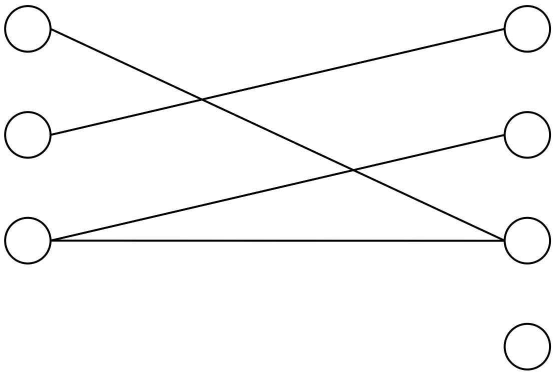


K_4

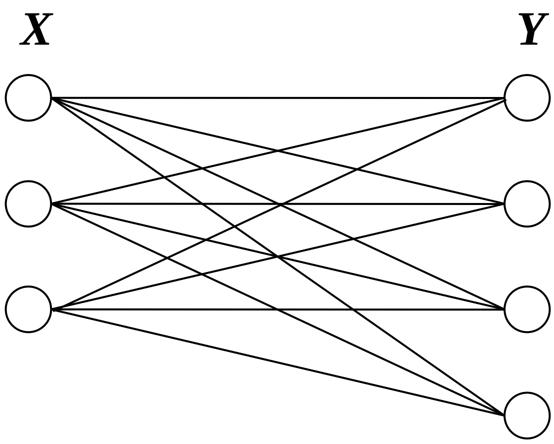
$\boldsymbol{K_n}$ **: the complete graph with** $\boldsymbol{n}$ **vertices**

Bipartite Graph : $V = X \cup Y$ $(X \cap Y = \varnothing)$

(*i*, *j*) (or $<i, j>$) : $i \in X,\ j \in Y$ **or** $i \in Y,\ j \in X$



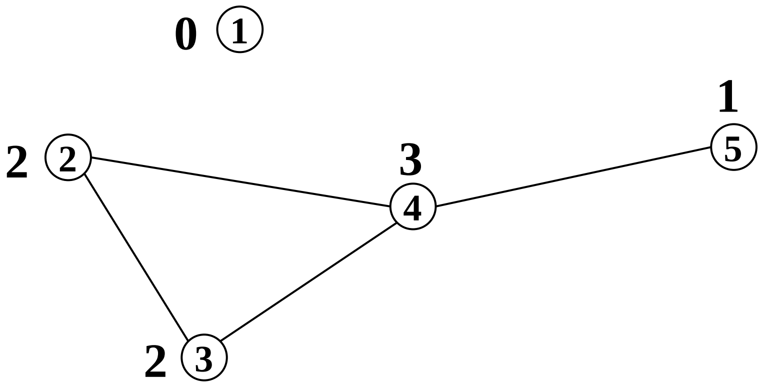
Complete Bipartite Graph : each vertex of *X* is connected to each vertex of *Y*



$K_{3,4}$

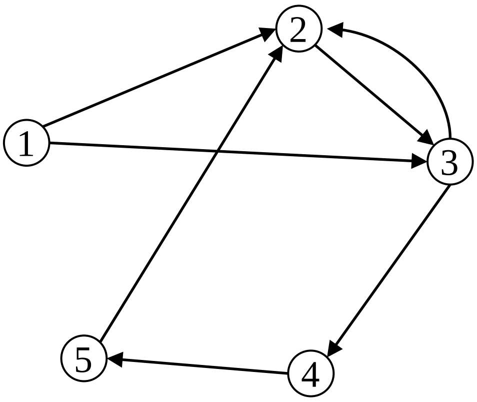
$K_{m,n}$ **: a complete bipartite graph with** $|X| = m$ **and** $|Y| = n$

Degree :



Let d_i be the degree of vertex i.

Theorem. $$\sum_{i \in V} d_i = 2 \cdot |E|.$$



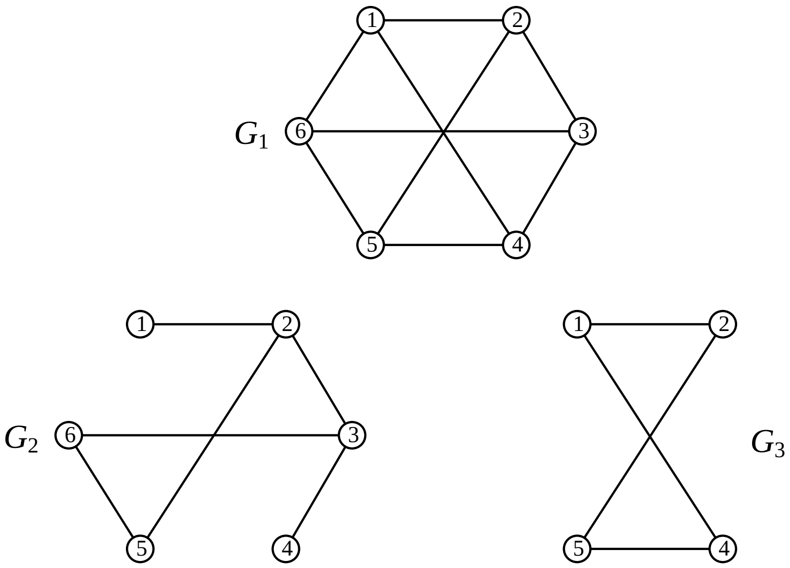
	indegree	outdegree
1	**0**	**2**
2	**3**	**1**
3	**2**	**2**
4	**1**	**1**
5	**1**	**1**

Let d_i^{in} and d_i^{out} denote the indegree and outdegree of vertex i.

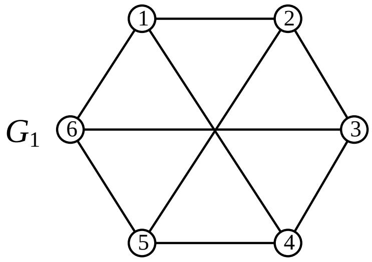
Theorem. $\sum_{i \in V} d_i^{in} = \sum_{i \in V} d_i^{out} = |E|$.

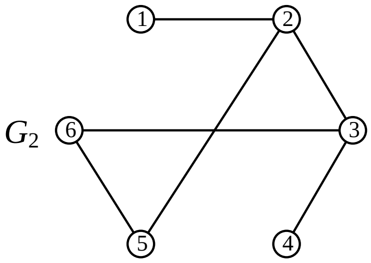
Subgraph : $G' = (V', E')$ is a subgraph of $G = (V, E)$

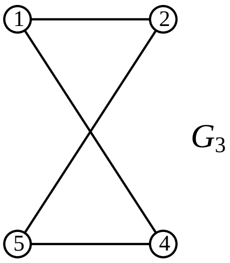
$$\Rightarrow \; V' \subseteq V \text{ and } E' \subseteq E$$



G_2 and G_3 are two subgraphs of G_1.







G_1 is regular (3-regular or regular of degree 3).

G_2 is a spanning subgraph of G_1.

G_3 is an induced subgraph of G_1 by {1, 2, 4, 5}

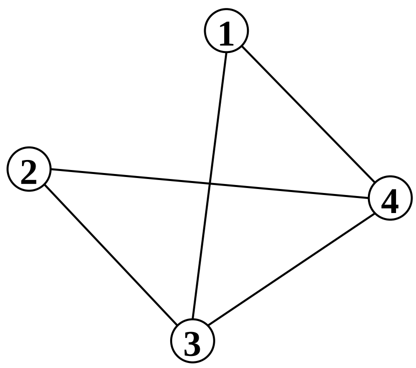
(or G_3 is a subgraph of G_1 induced by {1, 2, 4, 5}).

Isomorphism : $G_1 = (V_1, E_1)$ **and** $G_2 = (V_2, E_2)$ **are** ***isomorphic***

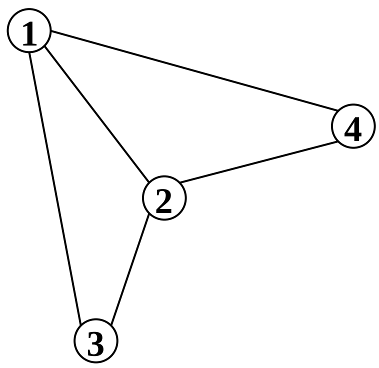
iff there exists a one-to-one and onto mapping

$f: V_1 \rightarrow V_2$ **such that** $(i, j) \in E_1 \Leftrightarrow (f(i), f(j)) \in E_2$.

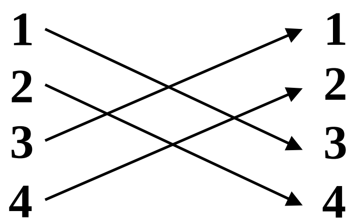
G_1



G_2

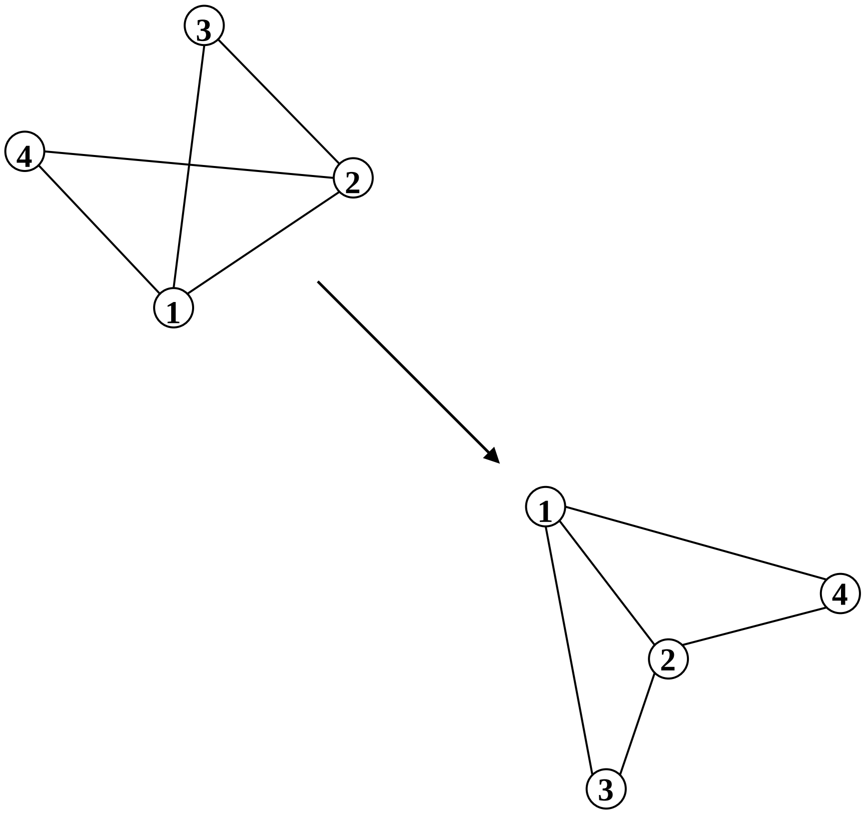


f

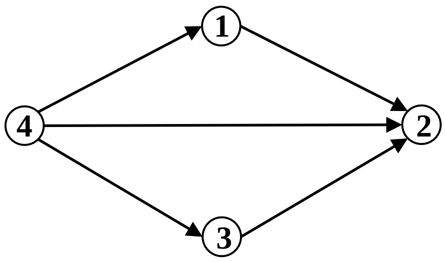


G_1 **and** G_2 **are isomorphic.**

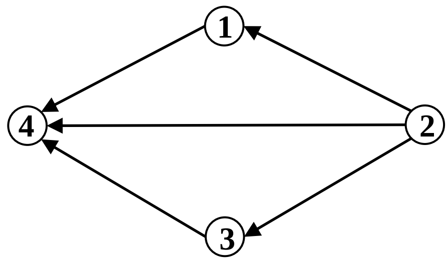
Relabel G_1 according to f :



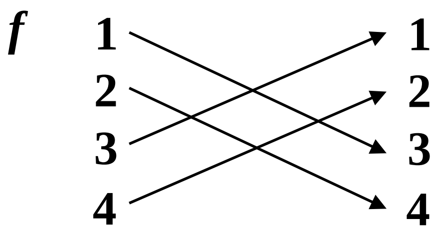
G_3



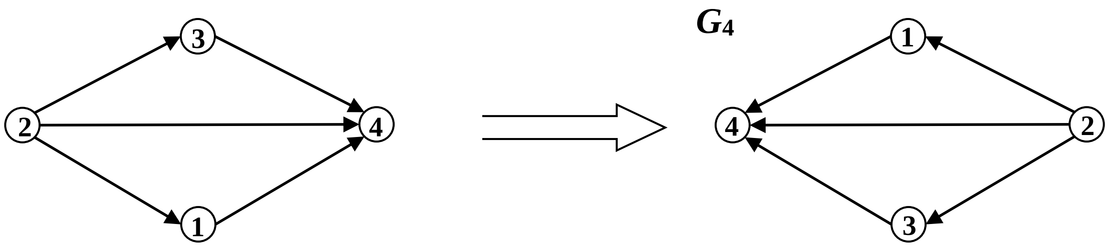
G_4

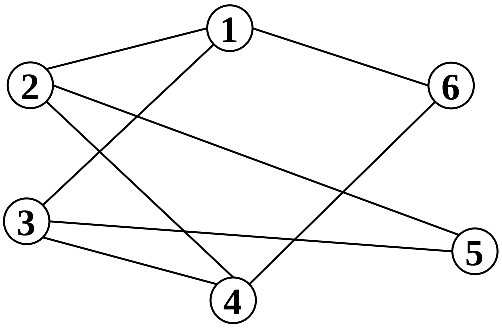


G_3 and G_4 are isomorphic.



Relabel G_3 according to f :





2, 1, 3, 5, 2, 1, 6 is a *walk*.

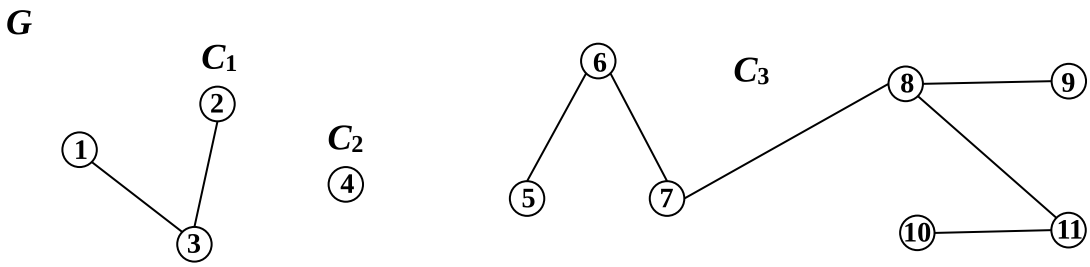
2, 1, 3, 4, 2, 5 is a *trail* (all edges are distinct) of length 5 from 2 to 5.

2, 1, 3, 4, 6 is a *path* (all vertices are distinct) of length 4 from 2 to 6.

3, 4, 2, 1, 3 is a *cycle* (all vertices are distinct) of length 4.

(3, 4, 2, 1, 3 is also a *circuit* (all edges are distinct) of length 4.)

Connected Components :



G consists of three connected components C_1, C_2, and C_3.

Connected Graphs :

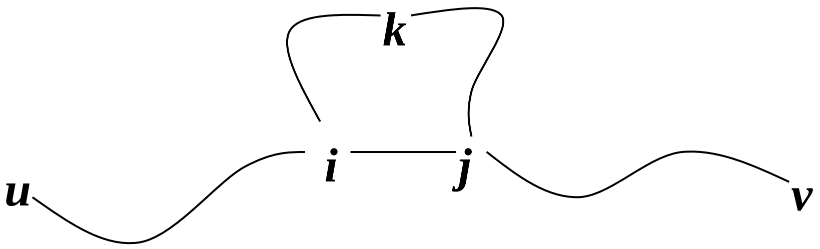
G is a *connected graph* iff it consists of one single connected component.

Theorem. **Let $G = (V, E)$ be a connected graph with $|V| > 1$. G contains either a vertex of degree 1 or a cycle (or both).**

Proof. **Consider that every vertex in G has degree >1. If we travel G from an arbitrary vertex, then a cycle will be formed after going through at most $|V|$ vertices.** □

Theorem. Let $G = (V, E)$ be a connected graph, and $(i, j) \in E$ be an edge that is contained in one cycle of G. Then, $G - (i, j)$ remains connected.

Proof.

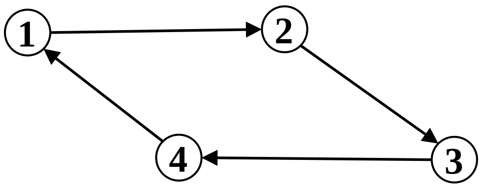


Let $u, \ldots, i, j, \ldots, v$ be a path from u to v that goes through (i, j). There exists another path $u, \ldots, i, \ldots, k, \ldots, j, \ldots, v$ from u to v that does not go through (i, j). $\square$

Every *n*-vertex connected undirected graph contains at least $n-1$ edges.

A tree of *n* vertices is an *n*-vertex connected undirected graph that contains exactly $n-1$ edges, where $n \geq 1$.

A digraph is *strongly connected* iff it contains directed paths both from *i* to *j* and from *j* to *i* for every pair of distinct vertices *i* and *j*.



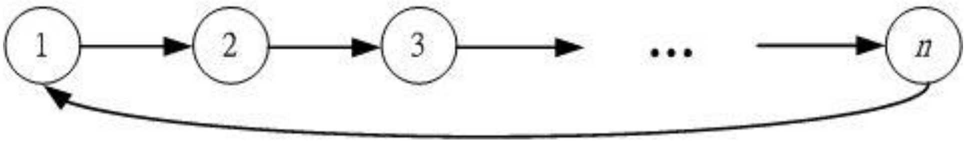
a strongly connected digraph

Theorem. Every *n*-vertex strongly connected digraph contains at least *n* arcs.

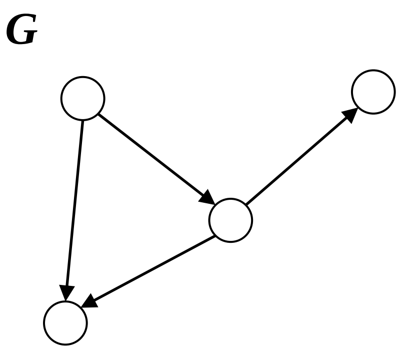
Proof. The definition of strongly connected digraphs assures $d_i^{in} \geq 1$ and $d_i^{out} \geq 1$ for every vertex *i*. Hence,

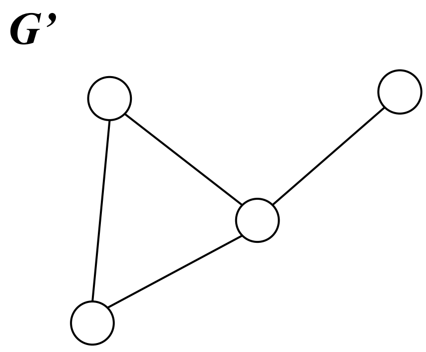
$$|E| = \sum_{i=1}^{n} d_i^{in} = \sum_{i=1}^{n} d_i^{out} \geq n. \qquad \square$$

There exists an *n*-vertex strongly connected digraph that contains exactly *n* arcs, where $n \geq 2$.



Underlying Graph :





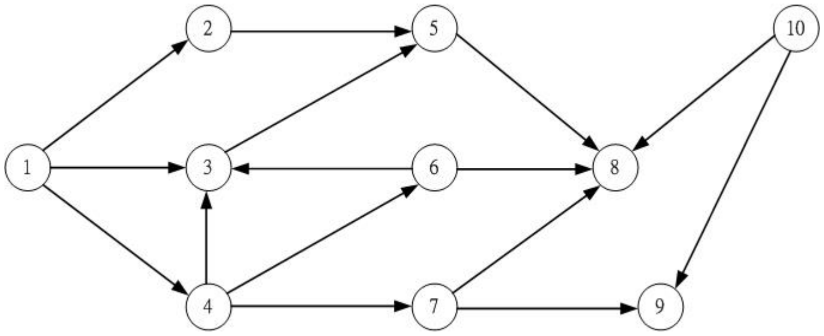
G'* is the underlying graph of *G

A digraph is *weakly connected* iff its underlying graph is connected.

(do Exercise #9)

Spanning Trees and Connectivity

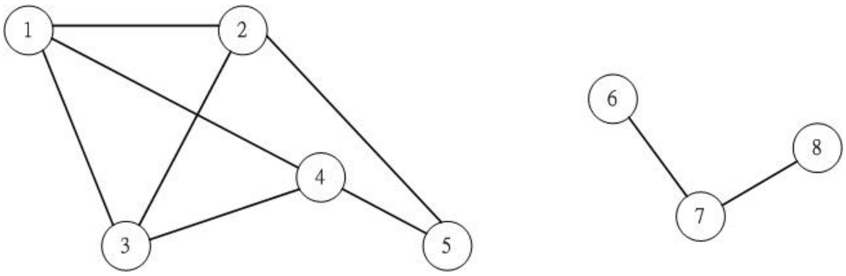
- **Breadth First Search (BFS)**



Start at vertex 1.

$\{1\} \Rightarrow \{2, 3, 4\} \Rightarrow \{5, 6, 7\} \Rightarrow \{8, 9\}$

Vertex 10 is not reachable from vertex 1.

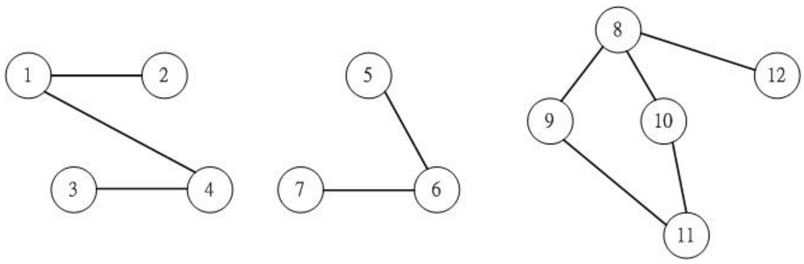


Start at vertex 1.

$\{1\} \Rightarrow \{2, 3, 4\} \Rightarrow \{5\}$

Vertices 6, 7, 8 are not reachable from vertex 1.

BFS can be used to determine the connected components of an undirected graph.



V **= {1, 2, 3, …, 12}**

Step 1. **Perform BFS starting at an arbitrary vertex (assume vertex 5).**

{5, 6, 7} is reachable from vertex 5.

⇒ {5, 6, 7} is a connected component.

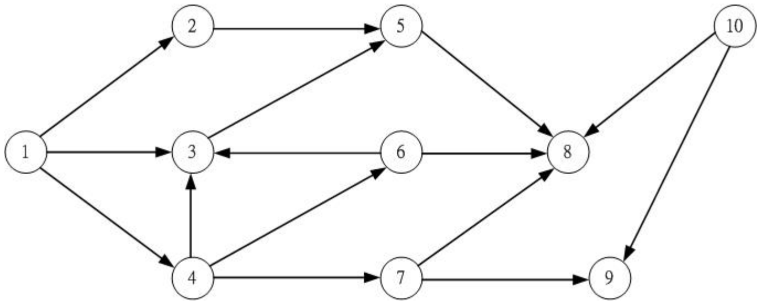
Step 2. **Repeat Step 1 for an arbitrary vertex (assume vertex 1) from the remaining vertices.**

{1, 2, 3, 4} is a connected component.

Step 3. **Repeat Step 1 for an arbitrary vertex from the remaining vertices.**

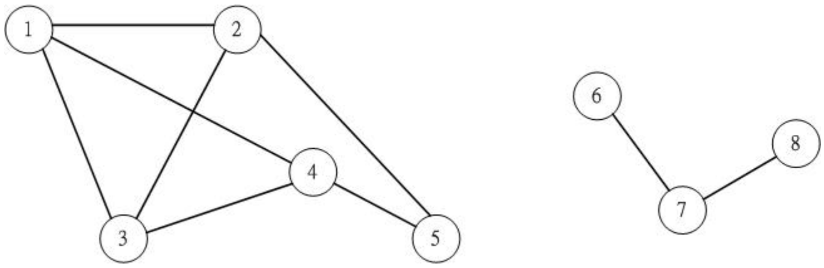
{8, 9, 10, 11, 12} is a connected component.

- **Depth First Search (DFS)**



Start at vertex 1.

$1 \rightarrow 4 \rightarrow 7 \rightarrow 9 \rightarrow 8 \rightarrow 6 \rightarrow 3 \rightarrow 5 \rightarrow 2$

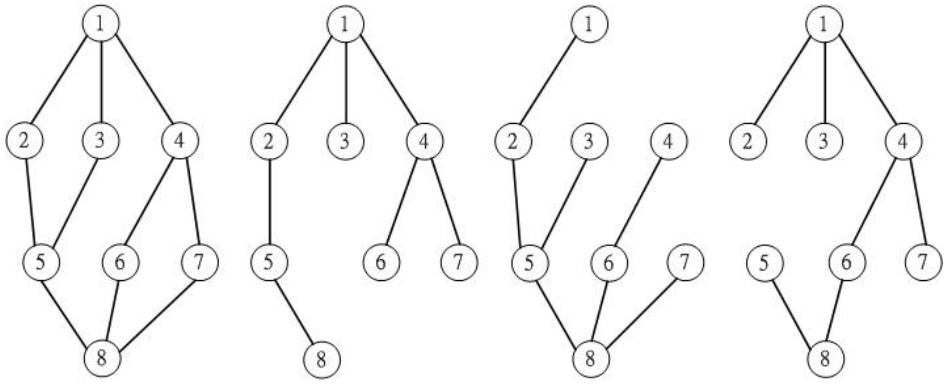


Start at vertex 1.

$$\mathbf{1 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 3}$$

DFS can be also used to determine the connected components of an undirected graph.

• Spanning Trees

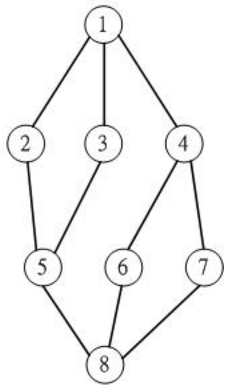


Let $G = (V, E)$ be an undirected graph.

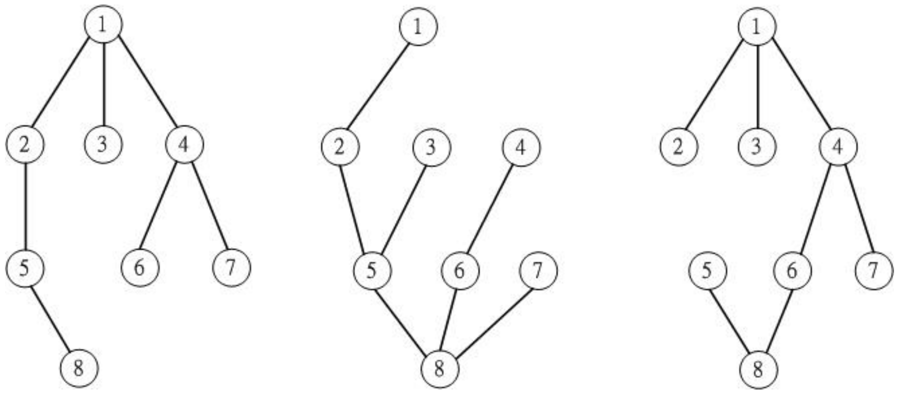
A subgraph $G_1 = (V_1, E_1)$ of G is a *spanning tree* of G iff $V_1 = V$ and G_1 is a tree.

Spanning trees of G can be obtained by BFS and DFS, named breadth-first spanning trees and depth-first spanning trees, respectively.

G:

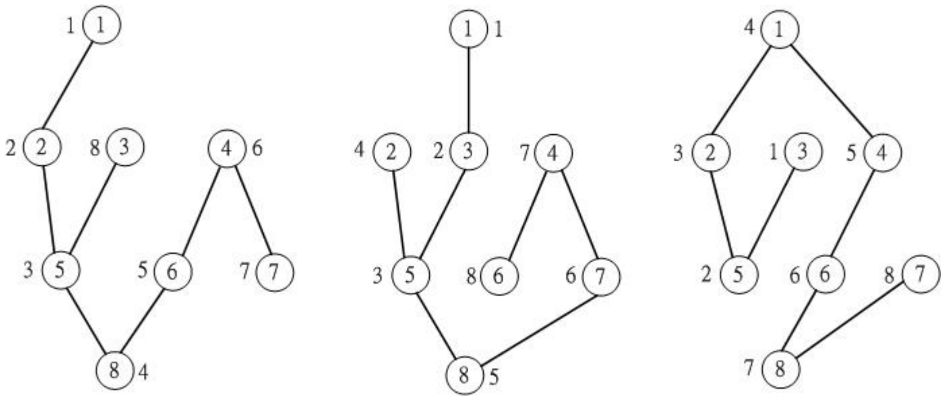


Some breadth-first spanning trees of *G* :



Root : 1 8 6

Some depth-first spanning trees of *G* :

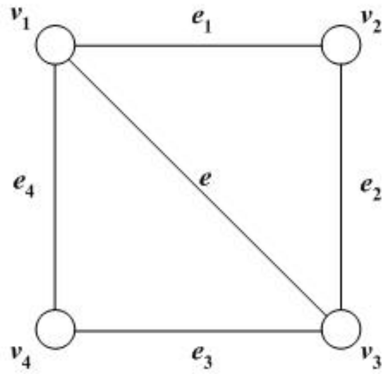


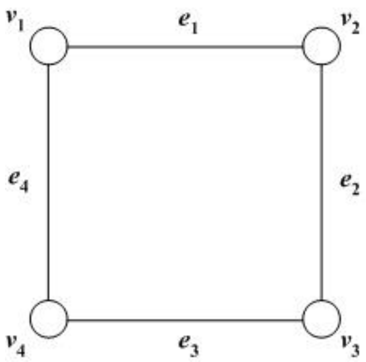
Root : 1 1 3

• Number of Spanning Trees

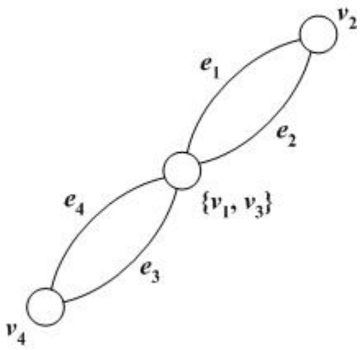
An edge *e* of *G* is said to be *contracted* if it is deleted and its both ends are identified; the resulting graph is denoted by $G \bullet e$.

***G* :**





$G - e$



$G \bullet e$

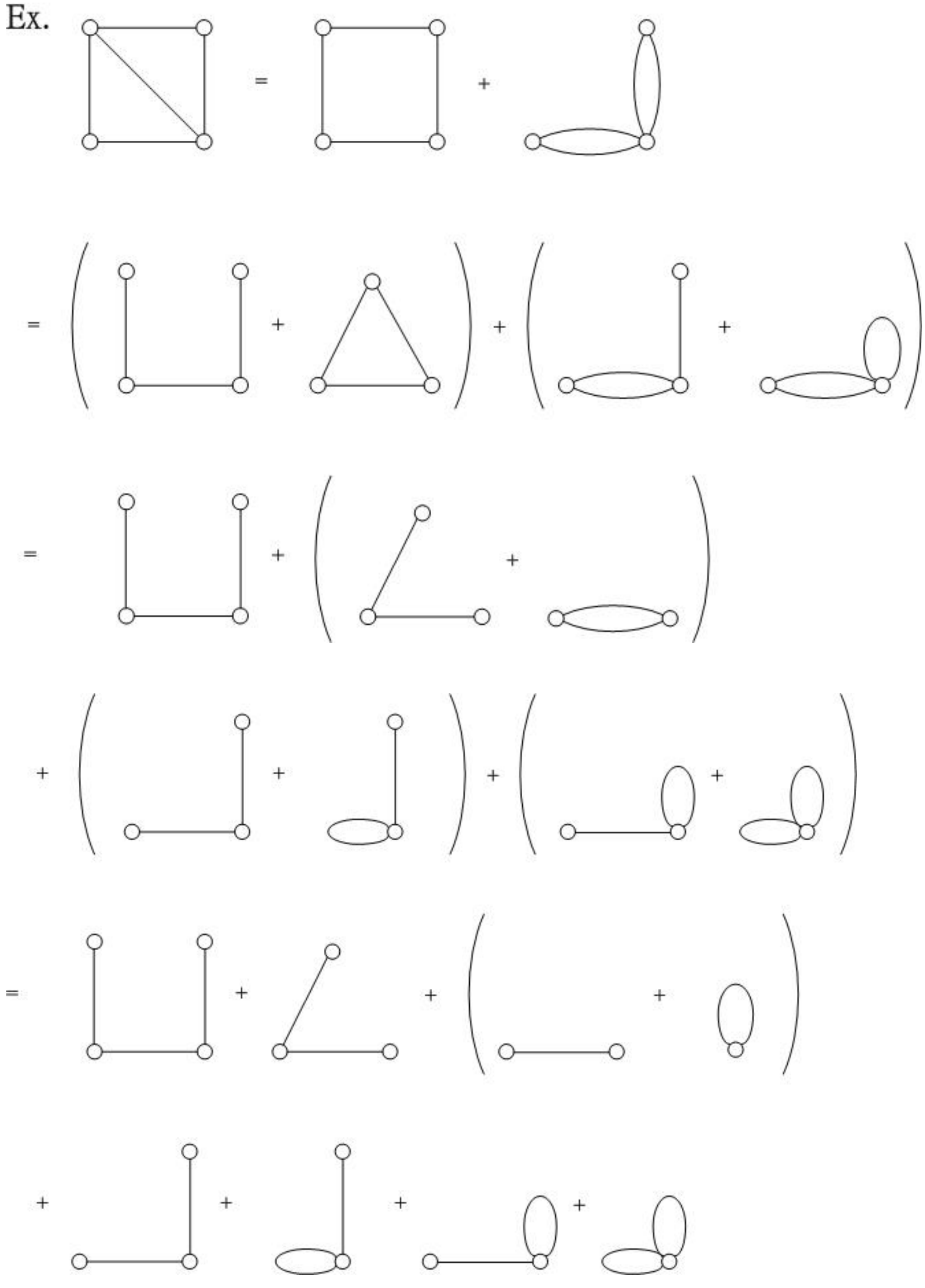
$S(G)$: the number of spanning trees of G.

$S(G-e)$: the number of spanning trees that do not contain edge e.

$S(G \bullet e)$: the number of spanning trees that contains edge e.

$$S(G) = S(G-e) + S(G \bullet e) \quad \text{for any edge } e \text{ of } G.$$

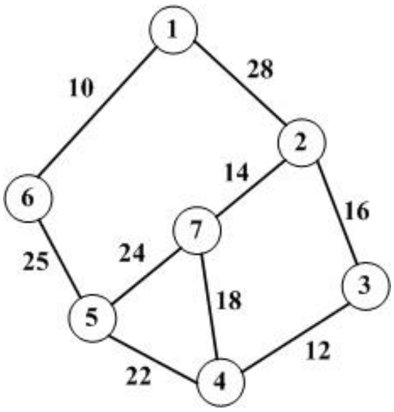
Ex.



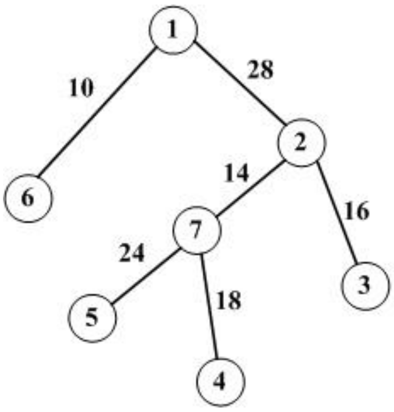
= 8.

- **Minimum (Cost) Spanning Trees (MSTs)**

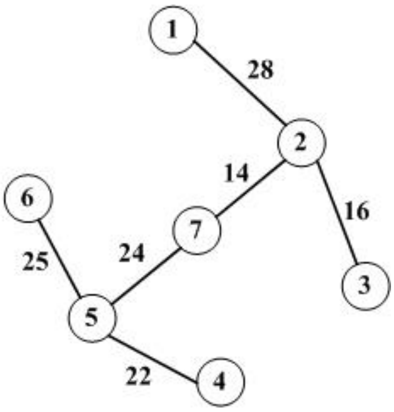
a weighted graph :

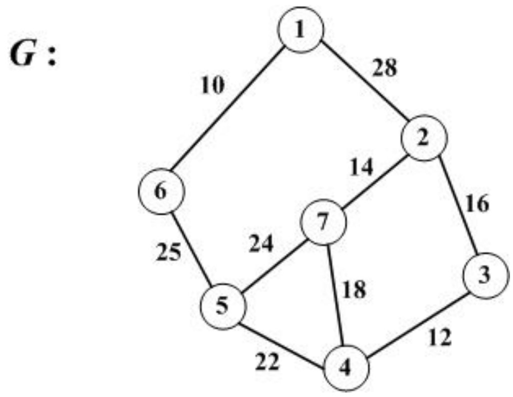


cost = 110



cost = 129



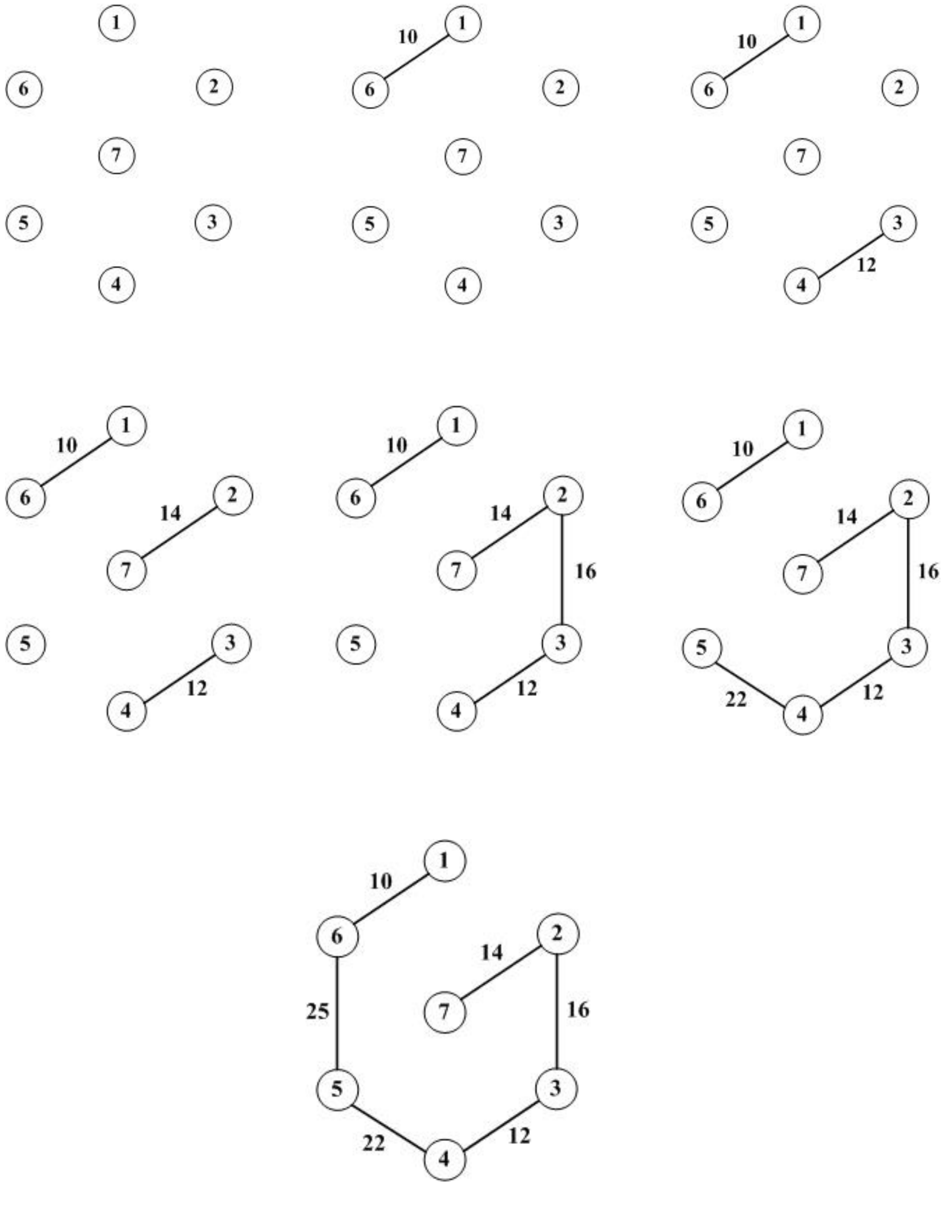


Kruskal's algorithm

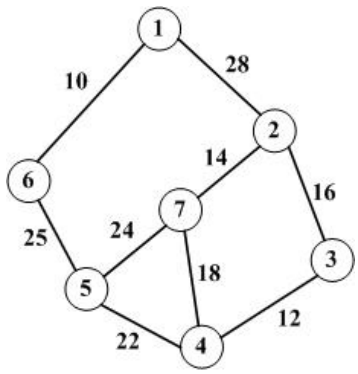
Step 1. **Sort edges nondecreasingly according to their costs.**

(1, 6), (3, 4), (2, 7), (2, 3), (4, 7), (4, 5), (5, 7), (5, 6), (1, 2)

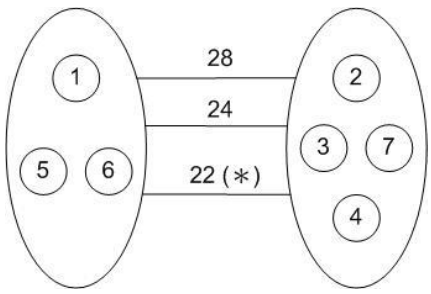
Step 2. **Examine the sorted sequence of edges sequentially. An edge is selected if including it does not cause a cycle.**

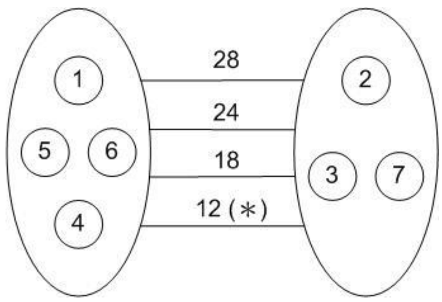


Prim’s algorithm

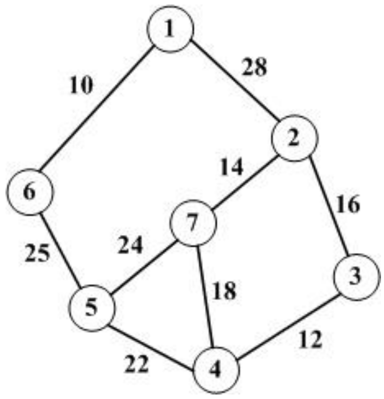


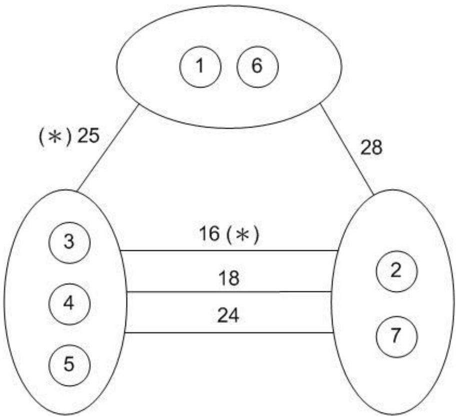
Assume that the starting vertex is 1.

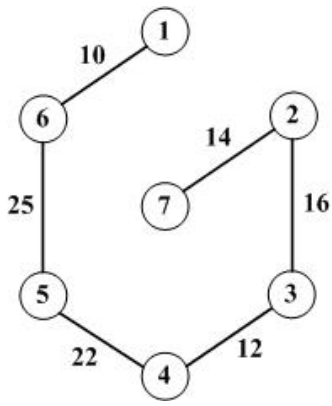




Sollin's algorithm







Theorem. Kruskal's algorithm, Prim's algorithm, and Sollin's algorithm each can generate an MST of *G*.

Correctness Proof of Kruskal's Algorithm :

We first assume that all edge costs are distinct.

Clearly, the output of Kruskal's algorithm is a spanning tree of *G*.

***T* : the spanning tree of *G* generated by Kruskal's algorithm.**

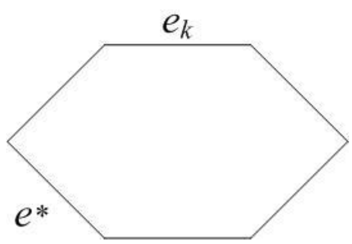
T : an MST of *G*.**

Suppose that *T* contains $e_1, e_2, \ldots, e_{n-1}$ and *T contains $e_1^*, e_2^*, \ldots, e_{n-1}^*$, both in increasing order of costs, where *n* is the number of vertices.**

Assume $e_1 = e_1^*$, $e_2 = e_2^*$, …, $e_{k-1} = e_{k-1}^*$, $e_k \neq e_k^*$, where $c(e_k) < c(e_k^*)$, as a consequence of Kruskal's algorithm.

After inserting e_k into T^*, a cycle is formed.

An edge, denoted by e^*, that is not in T can be found in this cycle.



Besides, $c(e^*) > c(e_k)$.

(If $c(e^*) < c(e_k)$ $(< c(e_k^*))$, then $e^* = e_r^* = e_r$ for some $1 \le r \le k-1$, a contradiction)

If e^* is replaced with e_k in T^*, then a spanning tree with smaller cost than T^* results, a contradiction.

If distinct edges may have the same cost, then

$c(e_k) \le c(e_k^*)$ **and** $c(e^*) \ge c(e_k)$.

(If $c(e^*) < c(e_k)$ $(\le c(e_k^*))$**, then** $e^* = e_r^* = e_r$ **for some**

$1 \le r \le k-1$**, a contradiction)**

When $c(e^*) > c(e_k)$**, replacing** e^* **with** e_k **in** T^* **will result in a spanning tree with smaller cost than** T^***, which is a contradiction.**

When $c(e^*) = c(e_k)$**, replacing** e^* **with** e_k **in** T^* **will result in another MST** T^{**} **with the same cost as** T^***.**

Then, repeat the above process on T^{**} **(hence** $e_k = e_k^*$**), and finally there is a contradiction or** $T = T^{**}$ **or another MST** T^{***} **with the same cost as** T^{**} **results.** □

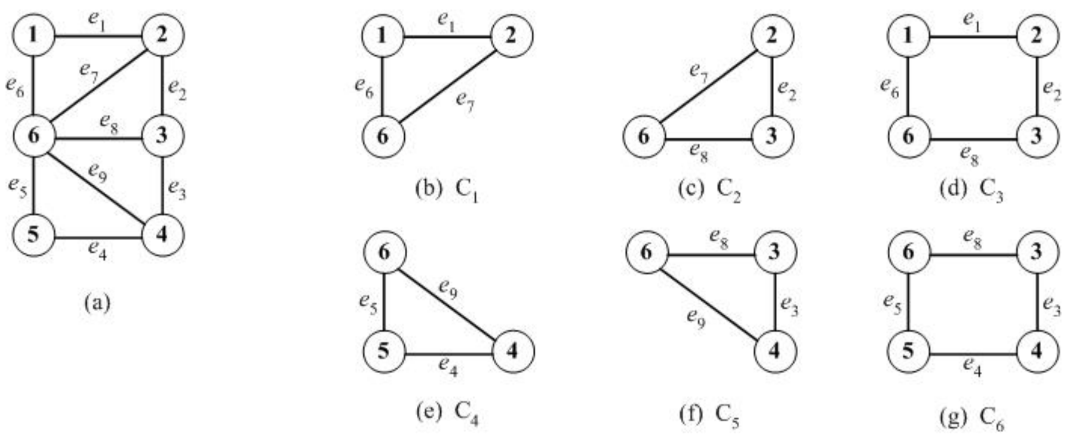
As a consequence of the proof above, G has a unique MST, if its edge costs are all distinct.

- # Cycle Basis

A, B : two sets.

Define $A \oplus B = A \cup B - A \cap B$. ($\oplus$: XOR)

Ex.



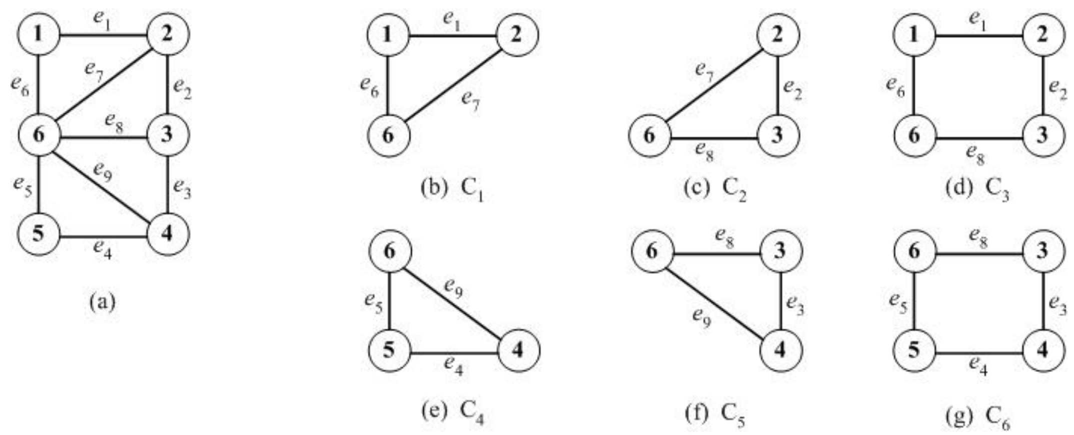
$c_1 = \{e_1, e_6, e_7\}$, $c_2 = \{e_2, e_7, e_8\}$

$\Rightarrow$ $c_1 \oplus c_2 = \{e_1, e_2, e_6, e_8\} = c_3$

$S = \{c_1, c_2, \ldots, c_k\}$ is a set of *independent cycles* iff

no cycle c_i in S is the XOR of some other cycles in S.

Ex. $\{c_1, c_2\}$ and $\{c_1, c_2, c_4, c_5\}$ are two sets of independent cycles.

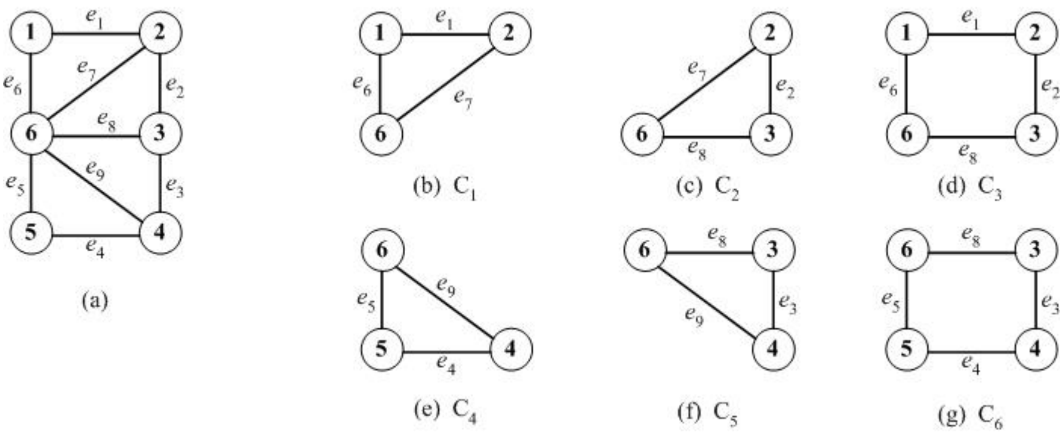


A set S of independent cycles in a graph G forms a *cycle basis* iff every cycle in G is the XOR of some cycles in S.

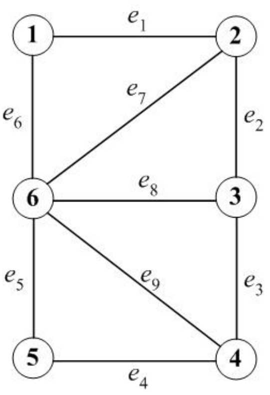
Ex. **$\{c_1, c_2, c_4, c_5\}$ forms a cycle basis, where**

$c_1 = \{e_1, e_6, e_7\}$, $c_2 = \{e_2, e_7, e_8\}$,

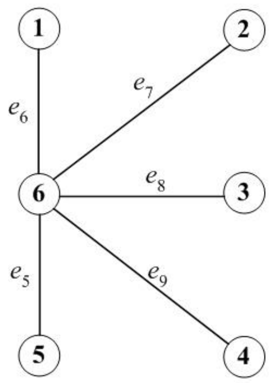
$c_4 = \{e_4, e_5, e_9\}$ **and** $c_5 = \{e_3, e_8, e_9\}$.

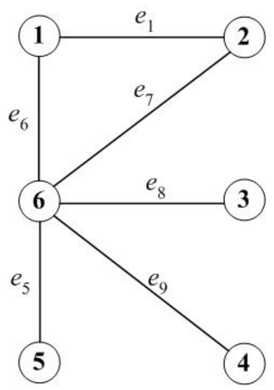


find a cycle basis of *G* :



Step 1. Find a spanning tree *T* of *G*.





Step 2. Create cycles by adding edges of *G* that are not in *T*.

Lemma. **The XOR of two distinct cycles is a cycle or a union of edge-disjoint cycles (i.e., one or more cycles can be found in the XOR of two distinct cycles).**

Proof : **Refer to :** ***Introduction to Combinatorial Mathematics*****, proof of Theorem 7-9, by Liu.**

Theorem. **Suppose**

T **: the set of edges in any spanning tree (or forest) of** $G = (V, E)$**;**

$E - T$ **:** $\{a_1, a_2, \ldots, a_k\}$**;**

c_i **: the unique cycle created by adding** a_i **to** T**.**

Then, $\{c_1, c_2, \ldots, c_k\}$ **is a cycle basis of** G**.**

Proof : **Each c_i contains a_i that is not in other cycles**

$\Rightarrow$ **$\{c_1, c_2, \ldots, c_k\}$ is a set of independent cycles.**

Let c**: a cycle of G;**

$c \cap (E - T) = \{a_{i_1}, a_{i_2}, \ldots, a_{i_r}\}$;

c_{i_j}**: the cycle created by adding a_{i_j} to T.**

Then, $c = c_{i_1} \oplus c_{i_2} \oplus \ldots \oplus c_{i_r}$, **for otherwise,**

according to the lemma (on page 48),

$c \oplus (c_{i_1} \oplus c_{i_2} \oplus \ldots \oplus c_{i_r})$ **should be a cycle or edge-disjoint cycles.**

There is a contradiction, because all the edges in $c \oplus (c_{i_1} \oplus c_{i_2} \oplus \ldots \oplus c_{i_r})$ **are contained in T.** □

• Connectivity

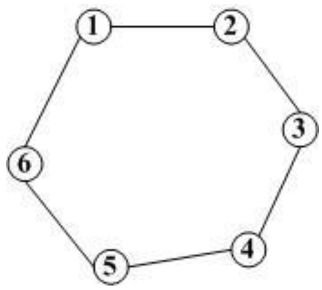
Let $G = (V, E)$ be a connected undirected graph.

A subset S of V is called a *vertex cut* of G iff

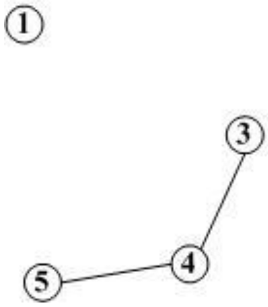
$G - S = (V - S, E - \{(i, j) \mid i \in S \text{ or } j \in S, (i, j) \in E\})$

is disconnected.

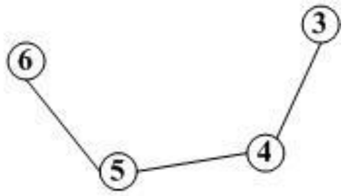
Ex.



{2, 6} is a vertex cut



{1, 2} is not a vertex cut



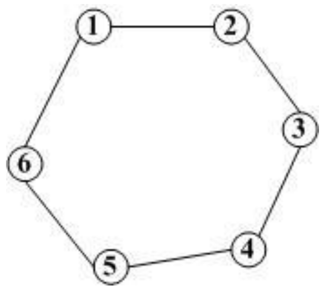
A *k-vertex cut* is a vertex cut of k vertices.

The (*vertex*) *connectivity* of G is the minimum k so that G has a k-vertex cut.

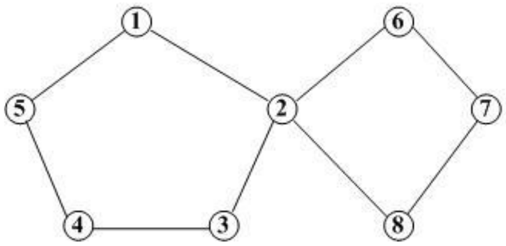
The connectivity of K_n, which has no vertex cut, is defined to be $n-1$.

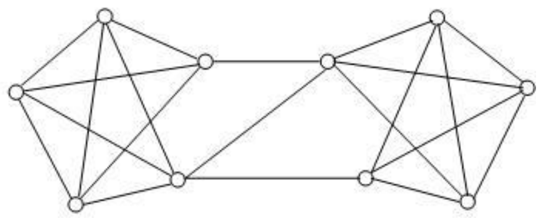
G is *k-connected*, if its connectivity $\geq k$.

Ex.



2-connected





1-connected **2-connected**

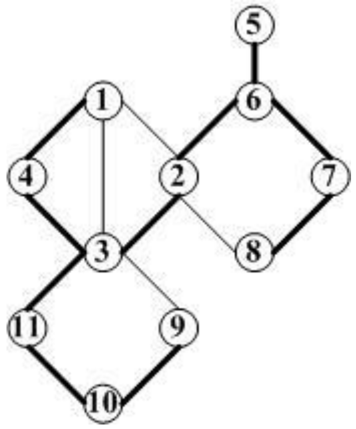
(vertex 2 is an articulation point)

v **is an *articulation point* of *G* iff {*v*} is a vertex cut of *G*.**

A connected graph *G* is *biconnected* (or 2-connected) iff *G* has no articulation point.

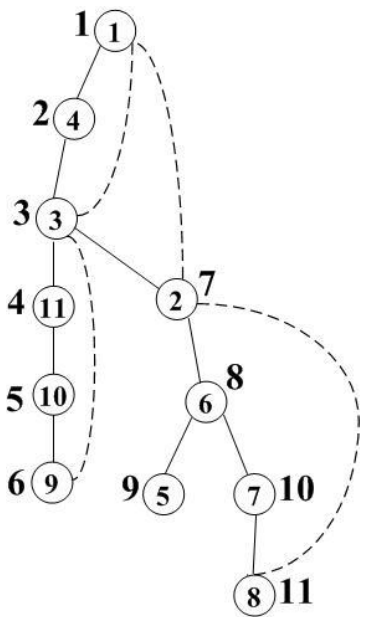
• Finding Articulation Points

A graph and one (rooted at ①) of its depth-first spanning trees :



redraw the spanning tree : —— **tree edges**

- - - - **back edges**



DFN **(Depth First Number) : the visiting sequence of vertices by DFS**

***L*(*i*) :** **the least *DFN* reachable from *i* through a path consisting of zero or more (downward) tree edges followed by zero or one back edge.**

vertex *i* :	**1**	**4**	**3**	**11**	**10**	**9**	**2**	**6**	**5**	**7**	**8**
***DFN*(*i*) :**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
***L*(*i*) :**	**1**	**1**	**1**	**3**	**3**	**3**	**1**	**7**	**9**	**7**	**7**
			Y				**Y**	**Y**			

(Y : an articulation point)

An edge (i, j) of G is a *cross edge* with respect to a spanning tree of G iff i is neither an ancestor nor a descendent of j.

Theorem. Suppose that G is a connected graph and T is a depth-first spanning tree of G. Then, G contains no cross edge with respect to T.

Proof. $(i, j) \in T \Rightarrow (i, j)$ **is not a cross edge.**

Consider $(i, j) \notin T$, and assume $DFN(i) < DFN(j)$.

$\Rightarrow$ **j belongs to the subtree of i**
(as a consequence of DFS)

$\Rightarrow$ **(i, j) is not a cross edge.** □

According to the theorem, the root of a spanning tree is an articulation point iff it has at least two children.

Theorem. Suppose that $G = (V, E)$ is a connected graph and T is a depth-first spanning tree of G. Then, $i \in V$ is an articulation point of G iff either

(a) i is the root of T and has at least two children, or

(b) i is not the root and has a child j with $L(j) \geq DFN(i)$.

Proof. We only prove (b).

j and its descendents will become isolated if i is removed. □

• Edge Connectivity

Given $G = (V, E)$, $S \subset E$ is an *edge cut* of G iff

$G - S = (V, E - S)$ is disconnected.

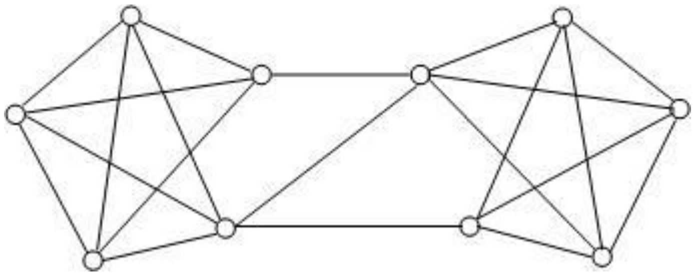
A *k-edge cut* is an edge cut of k edges.

The *edge connectivity* of G is the minimum k so that

G has a k-edge cut.

G is *k-edge-connected,* if its edge connectivity $\geq k$.

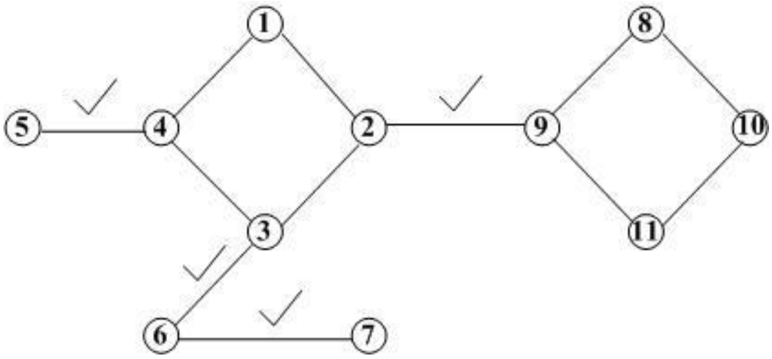
Ex.



3-edge-connected

(*i*, *j*) is a *bridge* of *G* iff {(*i*, *j*)} is an edge cut of *G*.

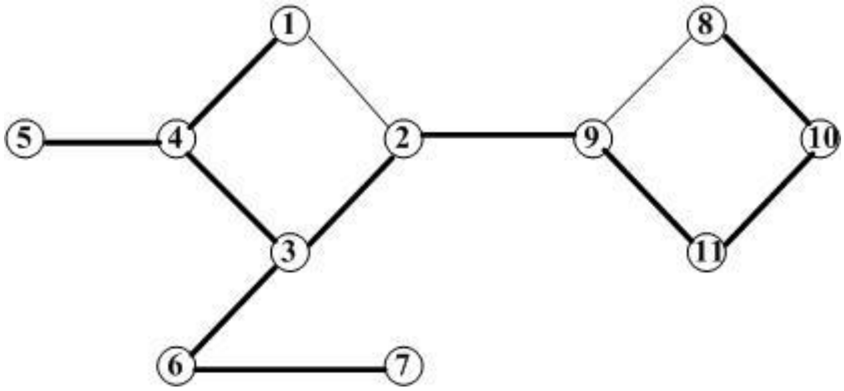
Ex.



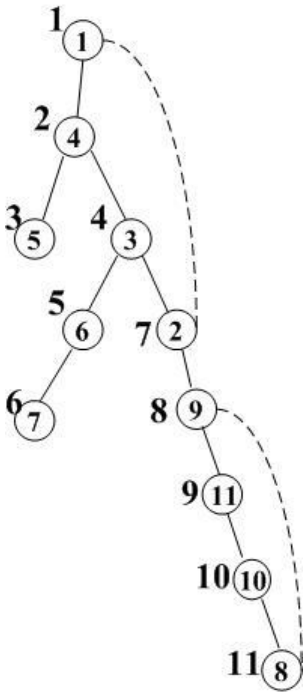
√ : bridges

• Finding Bridges

A graph and one (rooted at ①) of its depth-first spanning trees :



redraw the spanning tree :

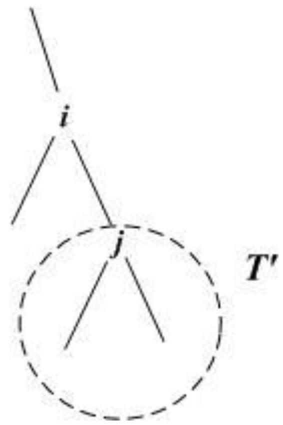


Vertex *i* :	**1**	**4**	**5**	**3**	**6**	**7**	**2**	**9**	**11**	**10**	**8**
***DFN*(*i*) :**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
***L*(*i*) :**	**1**	**1**	**3**	**1**	**5**	**6**	**1**	**8**	**8**	**8**	**8**
			Y		**Y**	**Y**		**Y**			

(Y : a bridge)

Theorem. **Suppose that** $G = (V, E)$ **is a connected graph. Let** $(i, j) \in E$, **and assume** $DFN(i) < DFN(j)$. **Then,** (i, j) **is a bridge of** G **iff** $L(j) = DFN(j)$.

Proof :



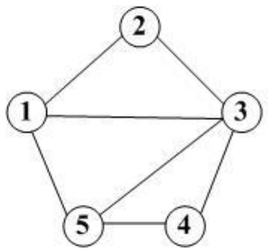
If $L(j) = DFN(j)$, **then** T' **will be isolated from** G **after removing** (i, j). □

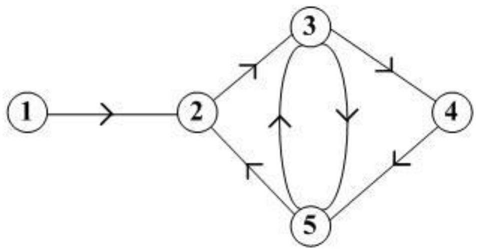
(do Exercise #10)

Paths

- ## Euler Trails, Euler Circuits

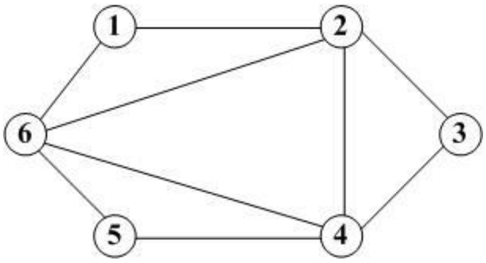
A trail (circuit) is called an *Euler trail* (*Euler circuit*) of *G* iff it traverses each edge of *G* exactly once.





Euler trail : 1, 2, 3, 1,
5, 3, 4, 5

Euler trail : 1, 2, 3, 5, 3
4, 5, 2



Euler circuit : 1, 2, 4, 3, 2, 6, 5, 4, 6, 1

Theorem. **Let $G = (V, E)$ be a connected undirected graph, where $|V| \geq 1$. Then,**

(1) **G has an Euler trail, but not an Euler circuit, iff it has exactly two vertices of odd degrees;**

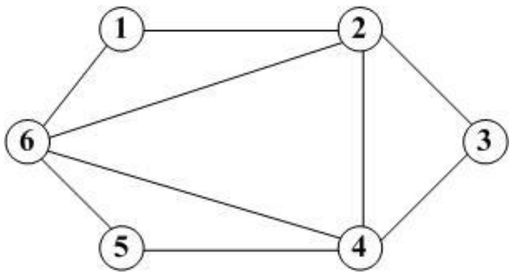
(2) **G has an Euler circuit iff all vertices have even degrees.**

Sketch of the proof.

($\Rightarrow$) trivial.

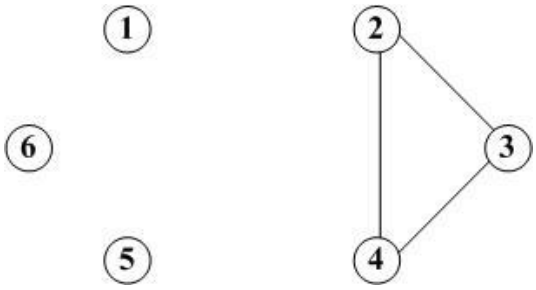
($\Leftarrow$) An Euler circuit for (2) can be constructed as follows

Step 1. **Start from any vertex, traverse an untraversed edge whenever it exists, and finally terminate at a vertex whose incident edges are all traversed.**



$$1 \to 2 \to 6 \to 5 \to 4 \to 6 \to 1$$

Step 2. **Remove traversed edges and repeat Step 1 for the remaining graph.**



$$2 \rightarrow 3 \rightarrow 4 \rightarrow 2$$

Step 3. **Expand the circuit obtained thus far with the circuit obtained by Step 2.**

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 6 \rightarrow 1$$

Step 4. **Repeat Step 2 and Step 3 until all the edges are traversed.**

An Euler trail for (1) can be obtained similarly, where the traversal starts from one odd-degree vertex and terminates at the other odd-degree vertex. □

Theorem. **Suppose that $G = (V, E)$ is a directed graph, where $|V| > 1$. Let d_i^{in} and d_i^{out} denote the indegree and outdegree of vertex i, respectively. Then, G has a u-to-v Euler trail iff the underlying graph of G is connected and either**

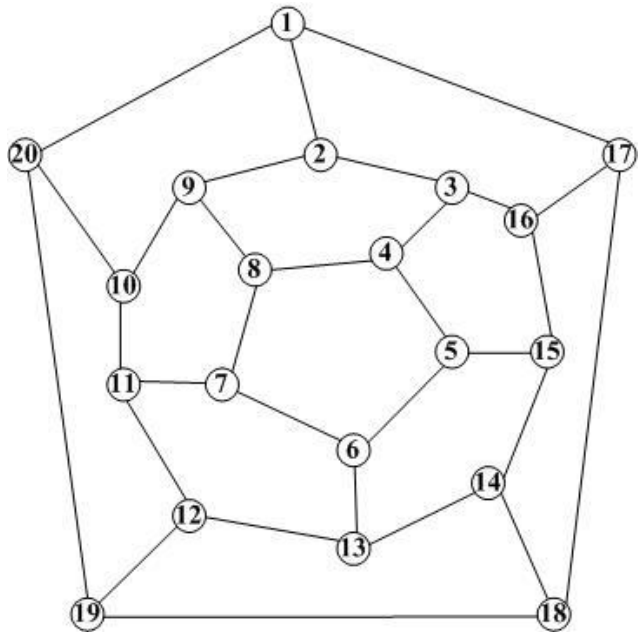
(1) $u = v$ and $d_i^{in} = d_i^{out}$ for every i in V, or

(2) $u \neq v$, $d_i^{in} = d_i^{out}$ for every i in $V - \{u, v\}$,
$d_u^{in} = d_u^{out} - 1$, and $d_v^{in} = d_v^{out} + 1$.

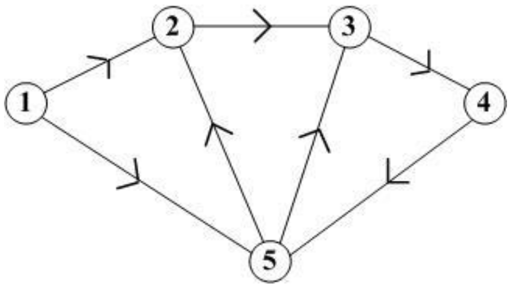
Proof. **Left as an exercise.**

- **Hamiltonian Paths, Hamiltonian Cycles**

A path (cycle) is called a *Hamiltonian path* (*cycle*) of *G* iff it goes through each vertex (exclusive of the starting vertex and ending vertex) of *G* exactly once.



a Hamiltonian cycle : 1, 2, 3, 4, …, 20, 1



A Hamiltonian path : 1, 2, 3, 4, 5

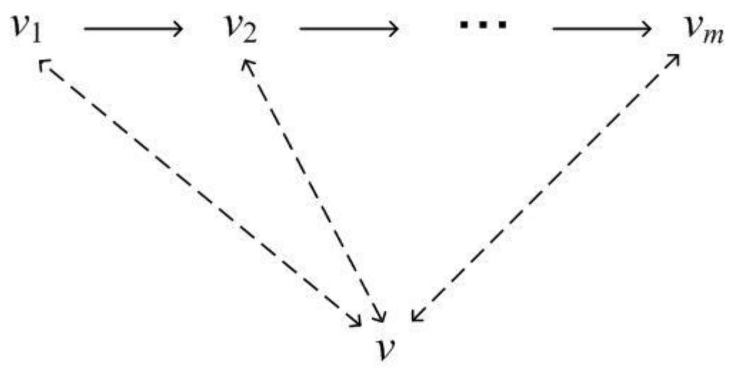
The problem of determining whether or not a graph has a Hamiltonian path (or cycle) is NP-complete.

There exist some sufficient (but not necessary) conditions for a graph having a Hamiltonian path or cycle.

Theorem. **Suppose that $G = (V, E)$ is a directed graph and between every two vertices u, v of G, there is one arc ($\langle u, v\rangle$ or $\langle v, u\rangle$). Then, there exists a directed Hamiltonian path in G.**

Proof. **Arbitrarily construct a directed path $\langle v_1, v_2\rangle$, $\langle v_2, v_3\rangle$, ..., $\langle v_{m-1}, v_m\rangle$ in G.**

Suppose that $m < |V|$ and v doesn't appear in the path.



If $<v, v_1> \in E$, then add $<v, v_1>$ to the path.

Otherwise, $<v_1, v> \in E$.

If $<v, v_2> \in E$, then add $<v_1, v>$ and $<v, v_2>$ to the path.

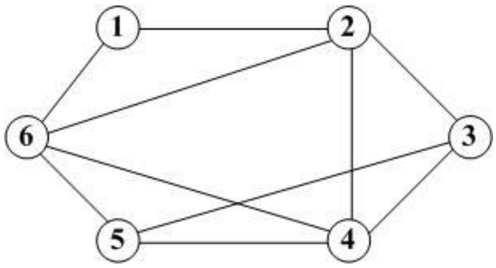
Otherwise, $<v_2, v> \in E$.

$$\vdots$$

Finally, either $<v_k, v>$ and $<v, v_{k+1}>$ are added to the path for some $1 \le k \le m-1$, or $<v_m, v>$ is added to the path. □

Theorem. Suppose that $G=(V, E)$ is an undirected graph, where $|V|=n\geq 2$. Let d_i be the degree of vertex v_i. If $d_i+d_j\geq n-1$ for every $(v_i, v_j)\notin E$ and $v_i\neq v_j$, then G has a Hamiltonian path.

Ex.



There exists a Hamiltonian path.

Proof. **First, $d_i + d_j \geq n - 1$ for every $(v_i, v_j) \notin E$ and $v_i \neq v_j$ can assure that G is connected, as explained below.**

Suppose to the contrary that G has two components C_1 and C_2 of n_1 vertices and n_2 vertices, respectively.

For every vertex v_1 of C_1 and every vertex v_2 of C_2,

$$d_1 + d_2 \leq (n_1 - 1) + (n_2 - 1) \leq n_1 + n_2 - 2 \leq n - 2,$$

which is a contradiction.

Then, a Hamiltonian path in G can be constructed as follows.

Step 1. **Arbitrarily construct a path** (v_1, v_2), (v_2, v_3), …, $(v_{m'-1}, v_{m'})$.

Step 2. **Extend the path by repeatedly adding a vertex** v **to the head (or tail) if** $(v, v_1) \in E$ **(or** $(v_{m'}, v) \in E$**).**

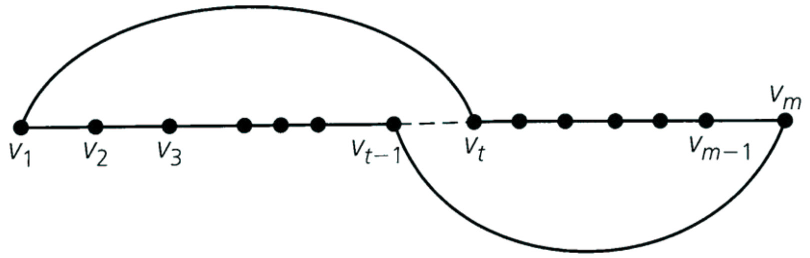
Suppose that the path (v_1, v_2), (v_2, v_3), …, (v_{m-1}, v_m) **results finally.**

If $m = n$, **then the path is Hamiltonian.**

Step 3. $(m \le n-1)$ **Construct a cycle on** $v_1, v_2, \ldots, v_m$.

If $(v_1, v_m) \in E$, **then a desired cycle is immediate.**

Otherwise $((v_1, v_m) \notin E)$, **select** (v_{t-1}, v_t) **such that** $(v_{t-1}, v_m) \in E$ **and** $(v_1, v_t) \in E$, **where** $3 \le t \le m-1$, **and obtain a desired cycle by deleting** (v_{t-1}, v_t).



The existence of (v_{t-1}, v_t) **is explained below.**

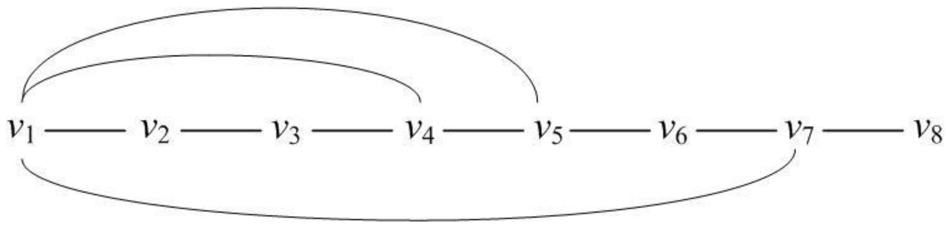
Notice that $d_1 \le m-2$ and $d_m \le m-2$ (because $(v_1, v_m) \notin E$)).

If no such (v_{t-1}, v_t) exists, then $d_m \le (m-2)-(d_1-1)$

("-1" excludes the case of $t=2$),

i.e., $d_1 + d_m \le m-1 < m$ ($\le n-1$), a contradiction.

For example, consider $m = 8$ ($\le n-1$) and $d_1 = 4$.



$d_1 \le 6$, $d_8 \le 6$ (because $(v_1, v_8) \notin E$))

$d_8 \le (m-2)-(d_1-1) \le 6-(4-1) = 6-3 = 3$

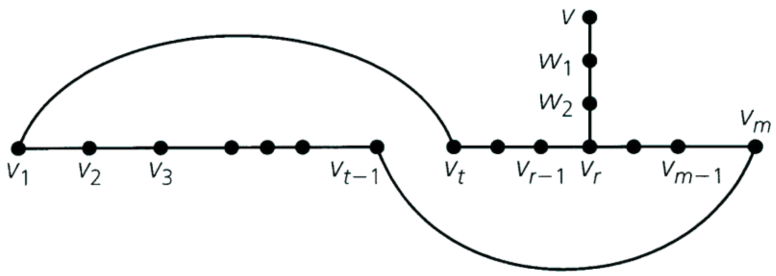
("6" means that v_8 has at most 6 neighbors $v_2, v_3, v_4, v_5, v_6, v_7$, and "$-3$" means that v_3, v_4, v_6 must be avoided, because $(v_{t-1}, v_8) \notin E$ whenever $(v_1, v_t) \in E$)

$\Rightarrow \quad d_1 + d_8 \le 4 + 3 = 7 < n-1$

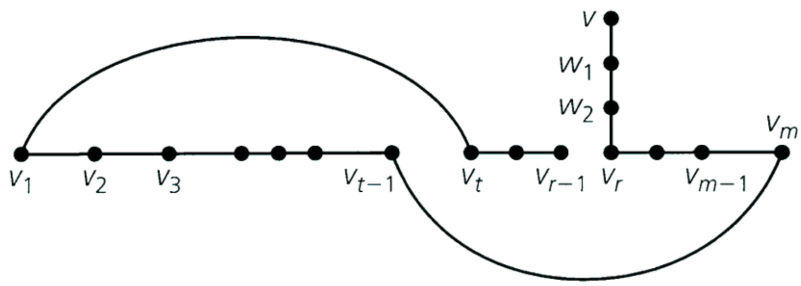
Step 4. Extend the cycle to a path of length > $m-1$.

Suppose $v \notin \{v_1, v_2, \ldots, v_m\}$.

Since G is connected, v is reachable to some vertex, say v_r, of the cycle.



A desired path can be obtained by removing (v_{r-1}, v_r) (or (v_1, v_t) if $r = t$).



Step 5. Repeat Step 2 to Step 4 until a Hamiltonian path results. □

Theorem. Suppose that $G = (V, E)$ is an undirected graph and $|V| = n \geq 3$. Let d_i be the degree of vertex v_i. If $d_i + d_j \geq n$ for every $(v_i, v_j) \notin E$ and $v_i \neq v_j$, then G has a Hamiltonian cycle.

Proof. Suppose that G has no Hamiltonian cycle.

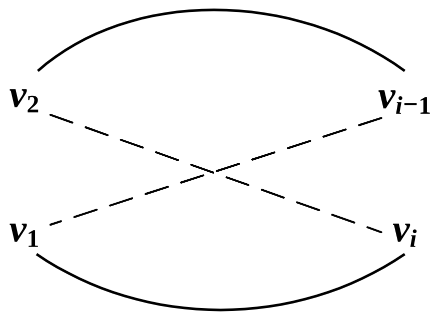
Augment G with a maximal number of edges so that the resulting graph, denoted by H, has no Hamiltonian cycle (i.e., any extra edge added to H will induce a Hamiltonian cycle in H).

Without loss of generality, assume that (v_1, v_2) is not an edge of H. If (v_1, v_2) is added to H, then there is a Hamiltonian cycle in H. Assume that the Hamiltonian cycle is as follows.

$$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow \ldots \rightarrow v_{i-1} \rightarrow v_i \rightarrow \ldots \rightarrow v_{n-1} \rightarrow v_n \rightarrow v_1$$

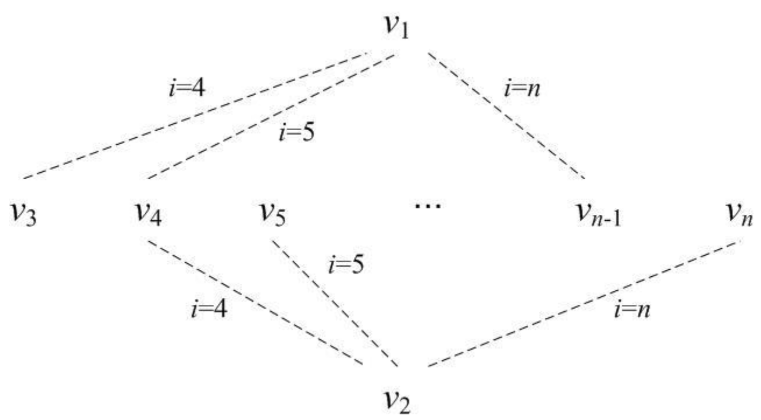
For each $3 \le i \le n$, (v_2, v_i) or (v_1, v_{i-1}) is not an edge of H, for otherwise H has a Hamiltonian cycle as follows.

$$v_2 \to v_i \to v_{i+1} \to \ \ldots \ \to v_{n-1} \to v_n \to v_1 \to v_{i-1}$$
$$\to v_{i-2} \to \ \ldots \ \to v_4 \to v_3 \to v_2$$



Therefore, $d_1 + d_2 \le n - 1$ (as explained below), which is a contradiction.

Consider the value of $d_1 + d_2$.



$i = 3$: $(v_2, v_3) \in E$ or $(v_2, v_3) \notin E$ (because $(v_1, v_2) \notin E$)

$\Rightarrow$ increase $d_1 + d_2$ by 0 or 1 (i.e., (v_2, v_3))

$4 \le i \le n$: $(v_1, v_{i-1}) \notin E$ or $(v_2, v_i) \notin E$

$\Rightarrow$ increase $d_1 + d_2$ by 0 or 1 (i.e., (v_1, v_{i-1}) or (v_2, v_i)) for each i

$\Rightarrow$ increase $d_1 + d_2$ by a total of $n - 3$ at most

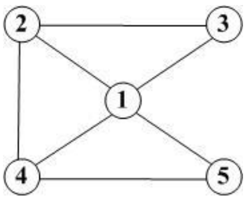
$(v_1, v_n) \in E$ or $(v_1, v_n) \notin E$

$\Rightarrow$ increase $d_1 + d_2$ by 0 or 1 □

Theorem. **Suppose that $G = (V, E)$ is an undirected graph and $|V| = n \geq 3$. If for every $1 \leq i \leq \lfloor (n-1)/2 \rfloor$, G has fewer than i vertices with degrees at most i, then G has a Hamiltonian cycle. When n is odd, G has a Hamiltonian cycle, even if G has $(n-1)/2$ vertices of degrees $(n-1)/2$.**

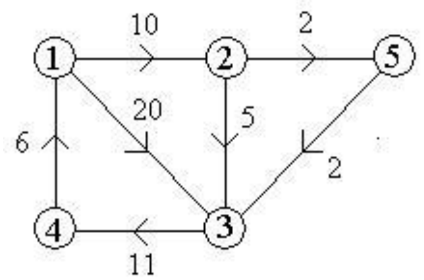
Proof : Refer to : R. Brualdi, *Introductory Combinatorics*, 1977.

Ex.



There exists a Hamiltonian cycle.

- **Shortest Paths**

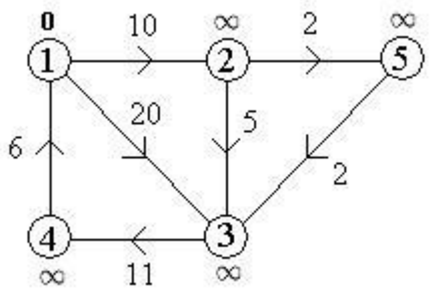


1-to-3 paths	**costs**	
1, 2, 3	**15**	
1, 2, 5, 3	**14**	**shortest**
1, 3	**20**	

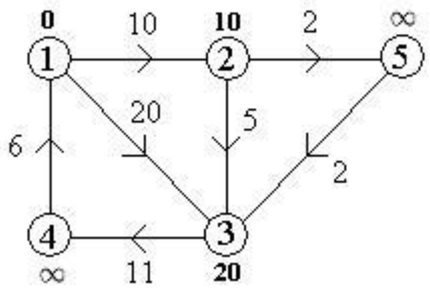
Dijkstra's shortest-path algorithm :

(choose the vertex with the minimum cost, repeatedly,

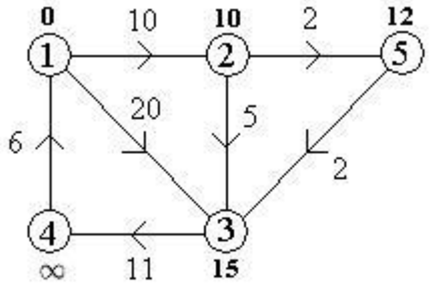
until all vertices are chosen)



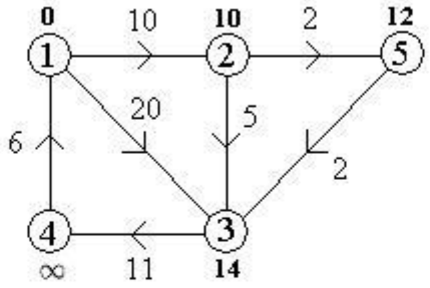
vertex 1 chosen



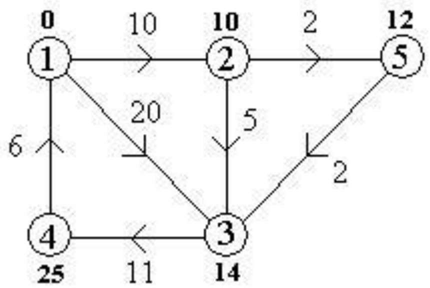
vertex 2 chosen



vertex 5 chosen



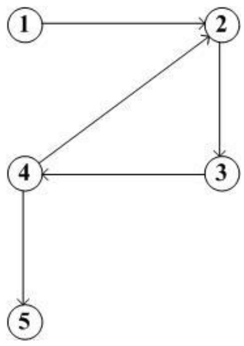
vertex 3 chosen



vertex 4 chosen

- **Transitive Closure**

(imagine a transitive relation of “reachability”)



adjacency matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$3 \xrightarrow{A(3,\,4)=1} 4 \xrightarrow{A(4,\,2)=1} 2$$

$$A^2(3,\,2)=1$$

$A^2(i, j) = 1$ **iff there exists an** i**-to-**j **walk of length 2.**

$$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A}^2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$A^3(i, j) = 1$ iff there exists an *i*-to-*j* walk of length 3.

$$\mathbf{A}^4 = \mathbf{A} \cdot \mathbf{A}^3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$A^4(i, j) = 1$ iff there exists an *i*-to-*j* walk of length 4.

The *transitive closure* of A is defined to be

$$A^+ = \sum_{i=1}^{\infty} A^i.$$

$A^+(i, j) = 1$ **iff there exists an i-to-j walk of length ≥ 1.**

When $i \neq j$, an i-to-j walk of length $\geq |V|$ implies an i-to-j walk of length $\leq |V| - 1$.

A longest cycle has length $|V|$.

Therefore,

$$A^+ = A + A^2 + A^3 + A^4 + A^5 \quad (|V| = 5)$$

$$= \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

If A is regarded as a relation, then A^+ is a transitive relation.

$$\text{Let } \mathbf{A}^0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The *reflexive transitive closure* of *A* is defined to be

$$A^* = \sum_{i=0}^{\infty} A^i = A^0 + A^1 + A^2 + A^3 + A^4 + A^5$$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

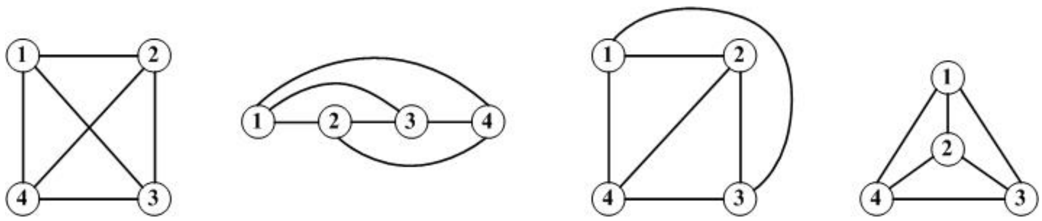
$A^*(i, j) = 1$ iff there exists an *i*-to-*j* walk of length ≥ 0.

(do Exercise # 11)

Miscellaneous Topics

• Planar Graphs

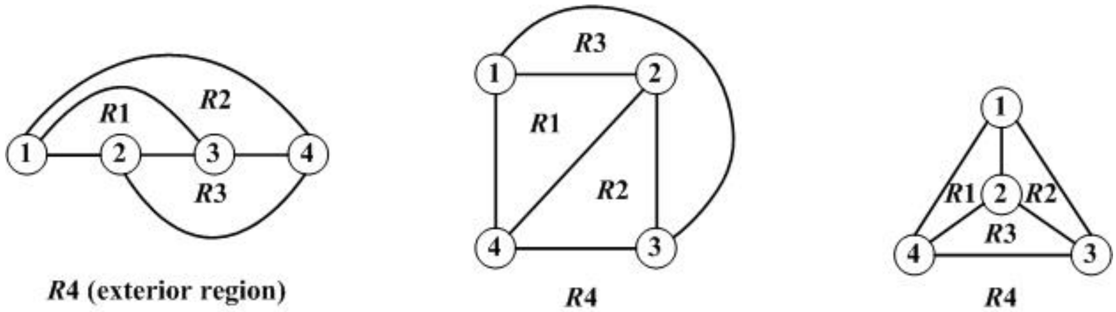
There are many ways to draw a graph.



A graph is *planar* iff it can be drawn so that no two edges cross.

Such a drawing is called a *planar drawing*.

A planar drawing partitions a planar graph into regions.



Theorem. **Let $G = (V, E)$ be a connected planar graph. Consider any planar drawing of G, and let r be the number of regions. Then,**

(1) $|V| - |E| + r = 2$;

(2) $|E| \le 3|V| - 6$, if $|E| \ge 2$.

Proof. Part (1). By induction on $|E|$.

Induction base.

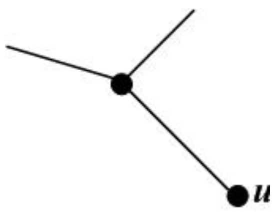
$$|E| = 1 \quad \Rightarrow \quad |V| = 2 \text{ and } r = 1$$
$$\Rightarrow \quad |V| - |E| + r = 2.$$

Induction hypothesis.

Assume $|V| - |E| + r = 2$ for $|E| = m \geq 1$.

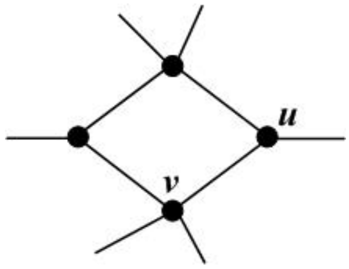
Induction step. Consider $|E| = m + 1$.

Case 1. G contains a vertex u with degree 1.



$$\text{remove } u \quad \Rightarrow \quad (|V| - 1) - (|E| - 1) + r = 2$$
$$\Rightarrow \quad |V| - |E| + r = 2$$

Case 2. *G* contains no vertex of degree 1.



$$\text{remove } (u, v) \Rightarrow |V| - (|E| - 1) + (r - 1) = 2$$

$$\Rightarrow |V| - |E| + r = 2$$

Part (2).

Case 1. $r = 1$.

$|E| = |V| - 1$ (G is a tree)

$|E| \geq 2 \Rightarrow |V| \geq 3$

Then, $(3|V| - 6) - |E| = (3|V| - 6) - (|V| - 1)$

$= 2|V| - 5$

≥ 1

Case 2. $r > 1$.

- every region is bounded by at least 3 edges
- every edge is shared by at most 2 regions

$\Rightarrow$ every region "consumes" at least 3/2 edges

$\Rightarrow$ $r \leq |E|/(3/2) = 2|E|/3$

$|V| - |E| + r = 2 \Rightarrow |V| - |E| + 2|E|/3 \geq 2$

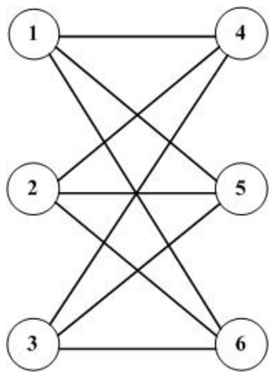
$\Rightarrow |E| \leq 3|V| - 6$ □

Corollary. **Every planar drawing of a connected planar graph has the same number ($r = 2 - |V| + |E|$) of regions.**

Corollary. **Suppose that $G = (V, E)$ is a planar graph with k connected components, and r is the number of regions in any planar drawing of G. Then, $|V| - |E| + r = k + 1$.**

Ex. **K_5 has $|V| = 5$ and $|E| = 10$. Since $|E| > 3|V| - 6$, K_5 is not planar.**

Ex. **$K_{3,3}$ satisfies $|E| \le 3|V| - 6$, but is not planar.**

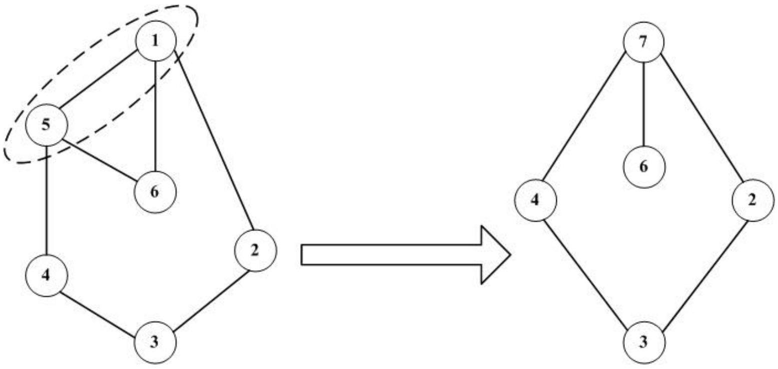


Suppose that *G* and *H* are two graphs.

***G* is an *elementary contraction* of *H* iff *G* can be obtained from *H* by replacing two adjacent vertices *i* and *j* of *H* by a single vertex *k*.**

All edges incident on either *i* or *j* (but not both) in *H* are now incident on *k*.

an elementary contraction



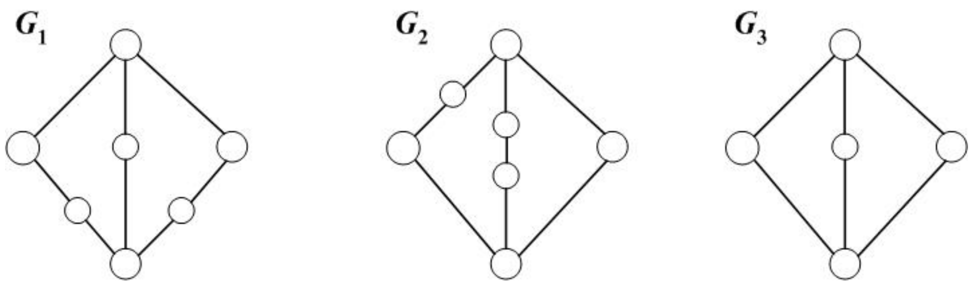
H ***G***

H **is** ***contractible*** **to** ***G*** **iff** ***G*** **can be obtained from** ***H*** **by a series of elementary contractions.**

Theorem. A graph ***G*** **is planar iff no subgraph of** ***G*** **is contractible to** $K_{3,3}$ **or** K_5**.**

Proof. Refer to : K. Wagner, "Über eine eigenschaft der ebenen komplexe," ***Math. Ann.*****, vol. 114, pp. 570-590, 1965, or J. A. Bondy and U. S. R. Murty,** ***Graph Theory with Applications*****, pp. 156, 1976.**

Two graphs are said to be *homeomorphic*, if they can be obtained from the same graph by adding vertices onto some of its edges, or one can be obtained from the other by the same way.



G_1 and G_2 are homeomorpic because they can be obtained from G_3 by adding vertices onto some of the edges of G_3.

G_1 and G_3 (G_2 and G_3) are homeomorphic, because G_1 (G_2) can be obtained from G_3 by adding vertices onto some edges of G_3.

Theorem. **A graph *G* is planar iff no subgraph of *G* is homeomorphic to K_5 or $K_{3,3}$.**

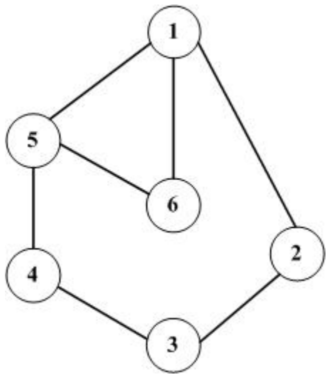
Proof. **Refer to : N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, 1974, or F. Harary, *Graph Theory*, 1972.**

• Matching

M $\subseteq$ ***E*** **is a *matching* in *G* = (*V*, *E*), if no two edges in *M* are incident on the same vertex.**

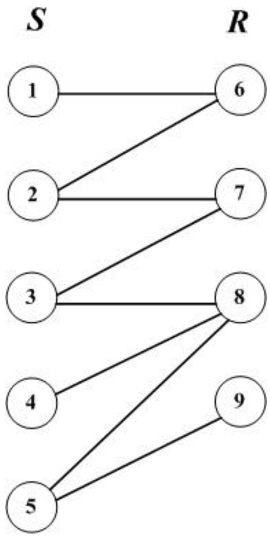
M **is a *maximum matching*, if there exists no matching *M'* with** $|M'| > |M|$ **in *G*.**

M **is a *perfect matching*, if** $|V| = 2 \times |M|$**.**



{(1, 6), (2, 3), (4, 5)} is both a maximum matching and a perfect matching.

Consider a bipartite graph $G = (V, E)$, where $V = R \cup S$.



Any matching M in G has $|M| \leq \min\{|S|, |R|\}$.

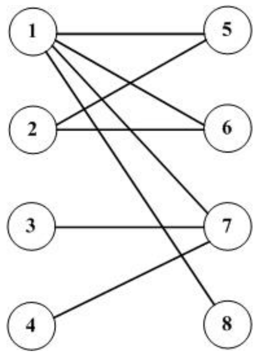
M is a *complete matching* iff $|M| = \min\{|S|, |R|\}$.

For example, $M = \{(2, 6), (3, 7), (4, 8), (5, 9)\}$ is a complete matching.

Theorem. **Suppose that $G = (R \cup S, E)$ is a bipartite graph, where $|R| \leq |S|$ is assumed. For any $W \subseteq R$, let ADJ(W) = $\{v \mid v$ is adjacent to a vertex in $W\}$. Then, G has a complete matching iff $|W| \leq |\mathrm{ADJ}(W)|$ for every $W \subseteq R$.**

If $|R| = |S|$, the iff-condition must be satisfied with every $W \subseteq R$ and every $W \subseteq S$.

Ex.



Let W = {3, 4}.

$\Rightarrow$ **ADJ(W) = {7}**

$\Rightarrow$ **$|W|$ > ADJ(W)**

$\Rightarrow$ **no complete matching**

Proof. $(\Rightarrow)$ **Clearly.**

$(\Leftarrow)$ **By induction on** $|R|$**.**

Induction basis. $|R| = 1$**.**

Induction hypothesis. Suppose that it holds for $|R| \leq m-1$**.**

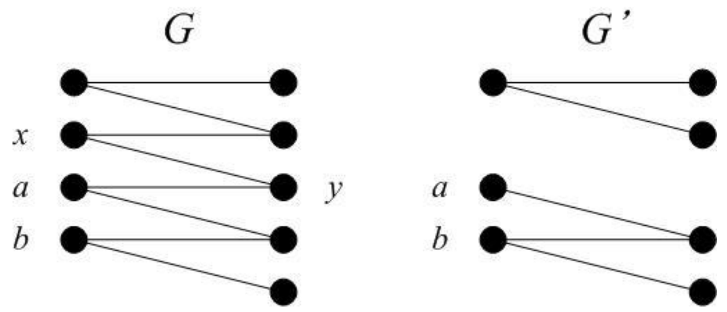
Then, three cases are considered for the situation of $|R| = m$**.**

- $|W| < |\mathrm{ADJ}(W)|$ **for every** $W \subseteq R$
- $|W| = |\mathrm{ADJ}(W)|$ **for** $W = R$ **and** $|W| < |\mathrm{ADJ}(W)|$ **for every** $W \subset R$
- $|W| = |\mathrm{ADJ}(W)|$ **for some** $W \subset R$

Case 1. $|W| < |\mathrm{ADJ}(W)|$ **for every** $W \subseteq R$.

Arbitrarily pick an edge (x, y), **where** $x \in R$ **and** $y \in S$.

Let G' **be the graph obtained from** G **by removing** x, y.



Clearly, if G' **has a complete matching, say** M',

then $M' + \{(x, y)\}$ **is a complete matching of** G.

In the following, we show that G' **has a complete matching.**

For every $W' \subseteq R - \{x\}$,

$|\mathrm{ADJ}(W')|$ in G' = $|\mathrm{ADJ}(W')|$ in G or

$|\mathrm{ADJ}(W')|$ in G' = $(|\mathrm{ADJ}(W')|$ in $G) - 1$.

(The latter holds when $y \in \mathrm{ADJ}(W')$ in G).

For example, $|\mathrm{ADJ}(\{a, b\})|$ in G' $(= 2)$

= $|\mathrm{ADJ}(\{a, b\})|$ in G $(= 3) - 1$.

Hence,

$|W'| <$ $|\mathrm{ADJ}(W')|$ in G (the assumption of this case)

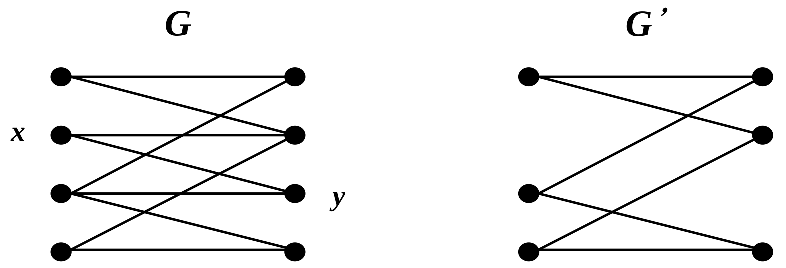
$\Rightarrow$ $|W'| \leq$ $|\mathrm{ADJ}(W')|$ in G'

$\Rightarrow$ G' has a complete matching

(by induction hypothesis)

Case 2. $|W| = |\mathrm{ADJ}(W)|$ **for** $W = R$ **and** $|W| < |\mathrm{ADJ}(W)|$

for every $W \subset R$**.**

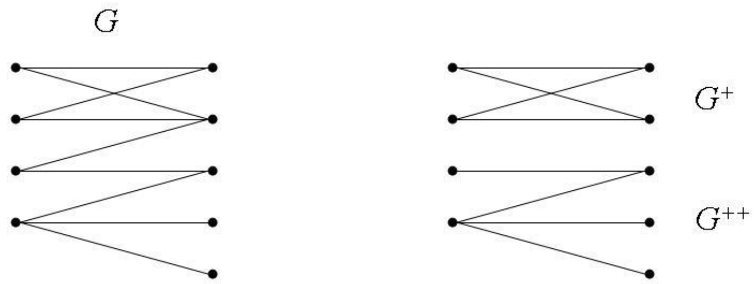


All the same as Case 1.

Case 3. $|W| = |\mathrm{ADJ}(W)|$ **for some** $W \subset R$.

G^+ : **the subgraph of** G **induced by** $W \cup \mathrm{ADJ}(W)$.

G^{++} : **the subgraph of** G **induced by** $(R - W) \cup (S - \mathrm{ADJ}(W))$.



If both G^+ **and** G^{++} **have complete matchings, then** G **has a complete matching (by combining the complete matchings of** G^+ **and** G^{++}**).**

Let $W' \subset W$.

$\mathrm{ADJ}(W')$ **in** G^+ **is identical with** $\mathrm{ADJ}(W')$ **in** G.

$|W'| \le |\mathrm{ADJ}(W')|$ **in** G **(the assumption of** G**)**

$\Rightarrow$ $|W'| \le |\mathrm{ADJ}(W')|$ **in** G^+

$\Rightarrow$ G^+ **has a complete matching (by induction hypothesis).**

For G^{++}, let $W'' \subseteq R - W$.

$|W''| \leq |\mathrm{ADJ}(W'')|$ in G^{++}, as explained below.

If $|W''| > |\mathrm{ADJ}(W'')|$, then (in G)

$$\begin{aligned} |W'' \cup W| &= |W''| + |W| \\ &> |\mathrm{ADJ}(W'')| + |\mathrm{ADJ}(W)| \\ &\geq |\mathrm{ADJ}(W'' \cup W)|, \end{aligned}$$

a contradiction to the assumption of G.

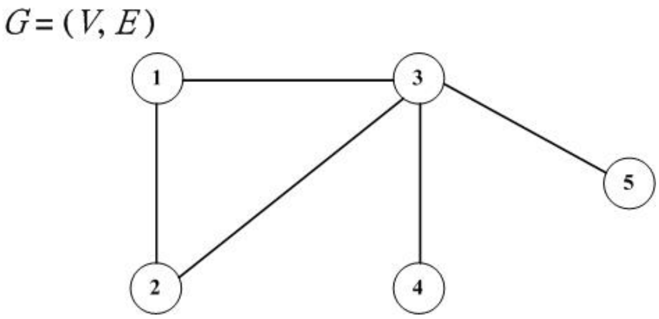
$\Rightarrow$ G^{++} has a complete matching

(by induction hypothesis).

□

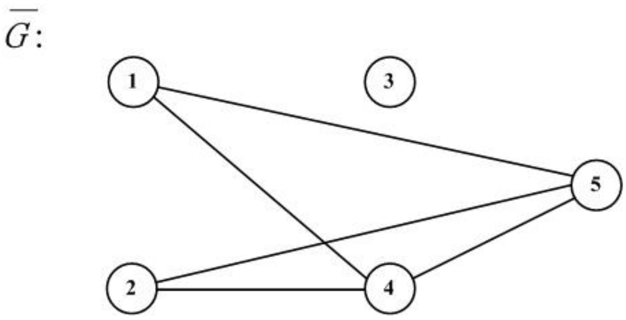
• Cliques, Independent Sets, Vertex Covers

A *clique* in a graph is a set of vertices every two of which are adjacent.



***V'* = {1, 2, 3} is a clique of size 3.**

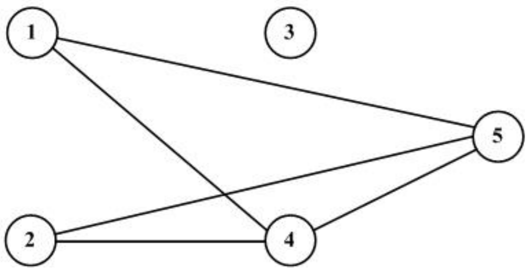
An *independent set* in a graph is a set of vertices no two of which are adjacent.



***V'* = {1, 2, 3} is an independent set of size 3.**

A *vertex cover* in a graph is a set of vertices such that each edge in the graph is incident with at least one vertex in the set.

$\overline{G}$:



***V* – *V'* = {4, 5} is a vertex cover of size 2.**

Theorem. **Suppose that $G = (V, E)$ is an undirected graph and $V' \subseteq V$. The following statements are equivalent.**

(1) V' is a clique of G.

(2) V' is an independent set of $\overline{G}$.

(3) $V - V'$ is a vertex cover of $\overline{G}$.

Proof. **(1) $\Leftrightarrow$ (2) $\Leftrightarrow$ (3).**

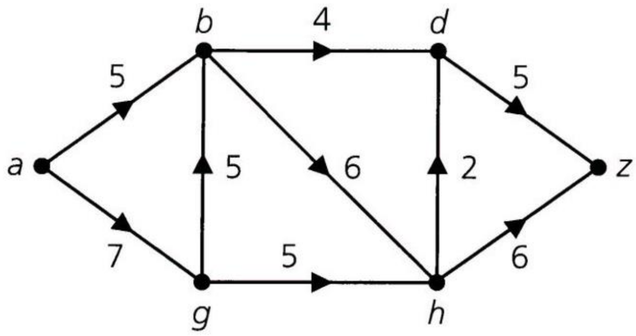
• Maximum Flow and Minimum Cut

Transport network $N = (V, E)$

(1) Weighted, connected, directed.

(2) A pair of *source* node a and *sink* node z.

(3) A nonnegative *capacity* $c(e)$ for each $e \in E$.



$$d_a^{in} = d_z^{out} = 0.$$

A *flow* is a function f from E to the set of nonnegative integers, satisfying

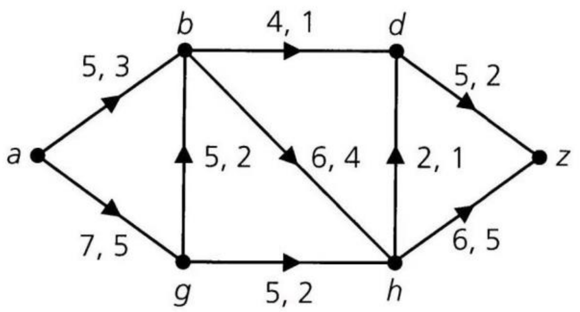
- **capacity constraint:** $0 \leq f(e) \leq c(e)$ **for each** $e \in E$;
- **conservation constraint:** $\psi^{+}(v) = \psi^{-}(v)$ **for** $v \notin \{a, z\}$,

where $\psi^{+}(v)$ $(\psi^{-}(v))$ **is the total flow into (out of)** v.

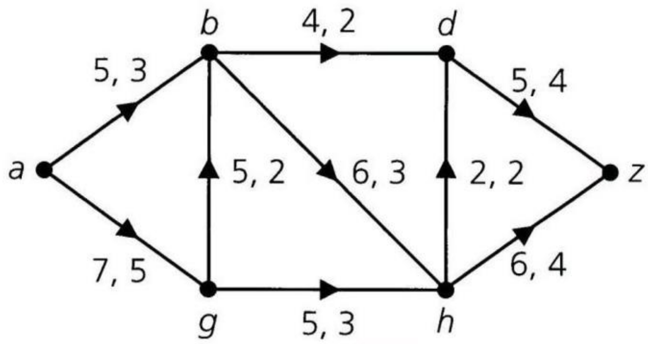
An arc e **is** ***saturated*** **if** $f(e) = c(e)$, **and** ***unsaturated*** **else.**

The ***total flow*** **(or** ***net flow*****) of** f **is defined to be**

$$F = \psi^{-}(a) = \psi^{+}(z).$$



f is not a flow, because $\psi^+(g) = 5 \neq 4 = \psi^-(g)$.



f is a flow, but $F = 8$ is not maximum.

$F \leq \min\{c(a, b) + c(a, g), c(d, z) + c(h, z)\} = \min\{12, 11\} = 11$.

The maximum flow problem is to determine f so that F is maximum.

Let $S \subset V$; $\bar{S} = V - S$;

$a \in S$; $z \in \bar{S}$;

$E(S; \bar{S})$ **be the set of arcs from** S **to** $\bar{S}$;

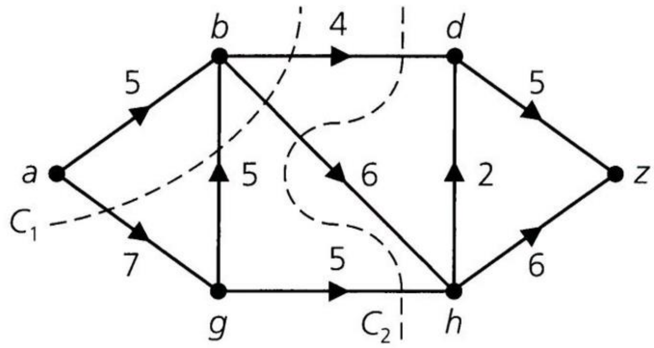
$E(\bar{S}; S)$ **be the set of arcs from** $\bar{S}$ **to** S.

$E(S; \bar{S}) \cup E(\bar{S}; S)$ **is called a *cut* (or *a-z cut*) of** N.

Define $c(S) = \sum_{e \in E(S;\bar{S})} c(e)$ **to be the *capacity* of the cut induced by** S.

$E(S; \bar{S}) \cup E(\bar{S}; S)$ **is a *minimum cut* if** $c(S)$ **is minimum.**

Ex.



(1) $S = \{a, b\}$, $\bar{S} = \{d, g, h, z\}$.

$E(S; \bar{S}) = \{\langle a, g\rangle, \langle b, d\rangle, \langle b, h\rangle\}$.

$E(\bar{S}; S) = \{\langle g, b\rangle\}$.

$\{\langle a, g\rangle, \langle b, d\rangle, \langle b, h\rangle, \langle g, b\rangle\}$ **is a cut.**

$c(S) = c(a, g) + c(b, d) + c(b, h) = 17$.

(2) $S = \{a, b, g\}$, $\bar{S} = \{d, h, z\}$.

$E(S; \bar{S}) = \{\langle b, d\rangle, \langle b, h\rangle, \langle g, h\rangle\}$.

$E(\bar{S}; S) = \varnothing$.

$\{\langle b, d\rangle, \langle b, h\rangle, \langle g, h\rangle\}$ **is a cut.**

$c(S) = c(b, d) + c(b, h) + c(g, h) = 15$.

Let $f(S, \bar{S}) \;=\; \sum_{<x,y>\in E(S;\bar{S})} f(x, y)$ **and**

$$f(\bar{S}, S) \;=\; \sum_{<y',x'>\in E(\bar{S};S)} f(y', x').$$

Lemma. (Conservation of Flow)

$F \;=\; f(S, \bar{S}) \;-\; f(\bar{S}, S)$ **for any $S \subset V$ and $a \in S$.**

Remarks.

(1) $F \leq c(S)$**.**

(2) $f(S, \bar{S}) \;-\; f(\bar{S}, S)$ **is the same (=F) for any S.**

Proof.

$$F = \sum_{v \in V} f(v, z) - \sum_{v' \in V} f(z, v'). \qquad \textbf{(1)}$$

For any $r \in \bar{S} - \{z\}$,

$$\psi^{+}(r) = \sum_{v \in V} f(v, r) = \sum_{v' \in V} f(r, v') = \psi^{-}(r), \qquad \textbf{(2)}$$

where $f(v, r) = 0$ $(f(r, v') = 0)$, if $(v, r) \notin E$ $((r, v') \notin E)$.

According to (1) and (2),

$$\begin{aligned} F &= \Big(\sum_{v \in V} f(v, z) - \sum_{v' \in V} f(z, v')\Big) + \\ &\quad \sum_{r \in \bar{S} - \{z\}} \Big(\sum_{v \in V} f(v, r) - \sum_{v' \in V} f(r, v')\Big). \qquad \textbf{(3)} \end{aligned}$$

$$\begin{aligned} &= \sum_{v \in V} f(v, \bar{S}) - \sum_{v' \in V} f(\bar{S}, v') \\ &= f(V, \bar{S}) - f(\bar{S}, V) \\ &= f(S \cup \bar{S}, \bar{S}) - f(\bar{S}, S \cup \bar{S}) \\ &= \big(f(S, \bar{S}) + f(\bar{S}, \bar{S})\big) - \big(f(\bar{S}, S) + f(\bar{S}, \bar{S})\big) \\ &= f(S, \bar{S}) - f(\bar{S}, S) \end{aligned}$$

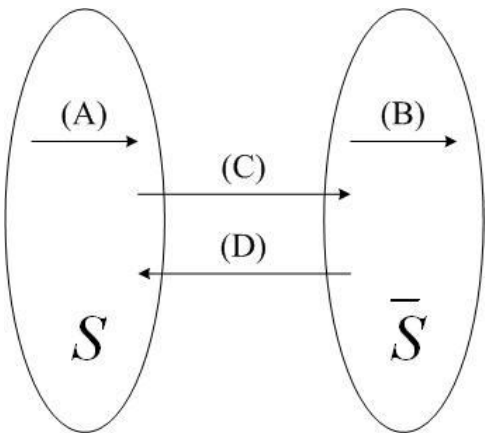
Another proof.

The right-hand side of (3) has the same value as

$$\underbrace{\sum_{<x,\, y> \in E(S;\, \bar{S})} f(x\,,\, y)}_{\textbf{(C)}} - \underbrace{\sum_{<y',\, x'> \in E(\bar{S};\, S)} f(y'\,,\, x')}_{\textbf{(D)}}, \qquad \textbf{(4)}$$

$$\Big(= \; f(S\,,\, \bar{S}) - f(\bar{S}\,,\, S) \Big)$$

as explained below.



Consider each arc <x, y> of N.

- $x \in S,\ y \in S \ \Rightarrow$ **$f(x, y)$ is not counted in (3), (4).**

  (A)

- $x \in \bar{S},\ y \in \bar{S} \ \Rightarrow$ **$f(x, y)$ is counted twice, positively and negatively, in (3), but not counted in (4).**

  (B)

- $x \in S,\ y \in \bar{S} \ \Rightarrow$ **$f(x, y)$ is counted once, positively, in (3), (4).**

  (C)

- $x \in \bar{S},\ y \in S \ \Rightarrow$ **$f(x, y)$ is counted once, negatively, in (3), (4).**

  (D)

Theorem. **If $F = c(S)$ for some $S \subset V$, then F is maximum and $c(S)$ is minimum.**

Proof. **According to Conservation of Flow,**

$$F' \leq c(S) = F \leq c(S')$$

for any total flow F' and any $(a \in)\, S' \subset V$, i.e., F is maximum and $c(S)$ is minimum.

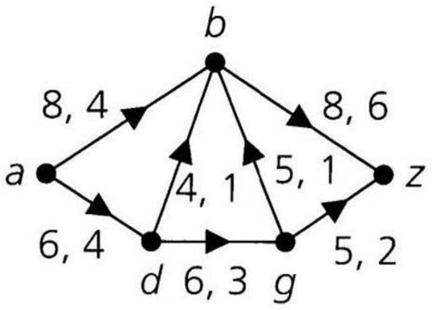
Theorem. $F = c(S)$ **if and only if**

(a) $f(e) = c(e)$ **for each** $e \in E(S;\overline{S})$;

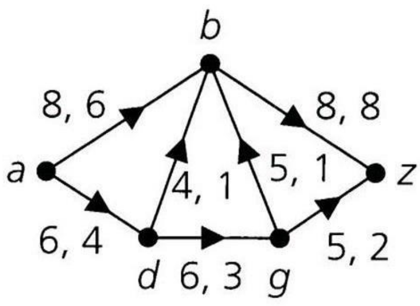
(b) $f(e) = 0$ **for each** $e \in E(\overline{S};S)$.

Proof. **A consequence of Conservation of Flow.**

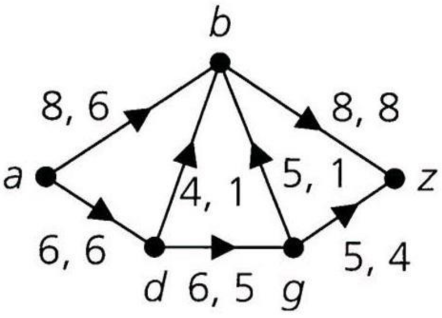
Ford & Fulkerson's algorithm:



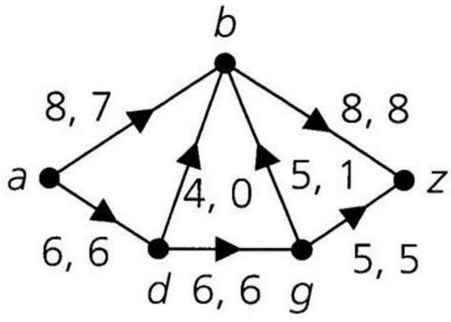
***F* = 8**



***F* = 10**
(augment (*a*, *b*, *z*) with 2)



***F* = 12**
(augment (*a*, *d*, *g*, *z*) with 2)



***F* = 13**
(augment (*a*, *b*, *d*, *g*, *z*) with 1)

Consider the augmenting path (*a*, *b*, *d*, *g*, *z*). Edges (*a*, *b*), (*d*, *g*) and (*g*, *z*) are *forward*, having the same direction as the flow, while (*b*, *d*) is *backward*, having the opposite direction to the flow.

A path in N is an *augmenting path*, if its each forward edge is unsaturated and its each backward edge e has $f(e) > 0$.

The maximal increment of flow by an augmenting a-to-z path P is equal to

$$\Delta_P = \min\{\min\{c(e) - f(e) \mid e \text{ is a forward edge}\}, \min\{f(e) \mid e \text{ is a backward edge}\}\}.$$

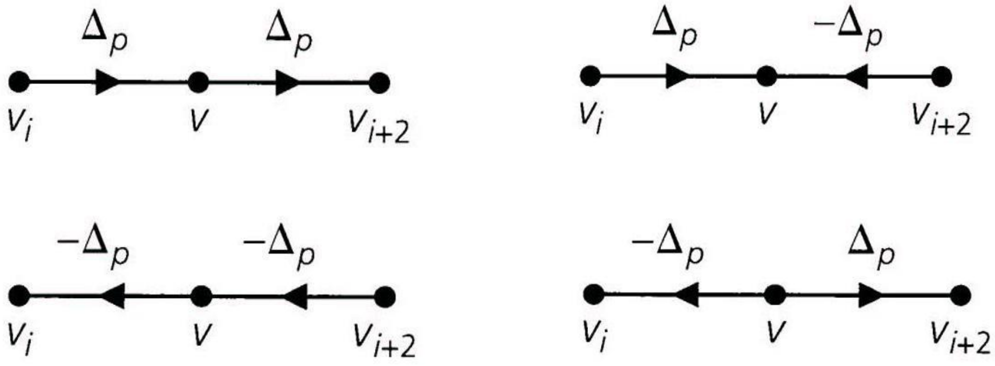
The correctness of Ford & Fulkerson's algorithm is shown below.

Theorem. **Suppose that f is a flow of N and P is an augmenting a-to-z path. Define f^+ as follows:**

$$f^+(e) = \begin{cases} f(e) + \Delta_P, & \text{if } e \text{ is a forward edge;} \\ f(e) - \Delta_P, & \text{if } e \text{ is a backward edge;} \\ f(e), & \text{if } e \text{ is not an edge of } P. \end{cases}$$

Then, f^+ is also a flow of N. The total flow increases by Δ_P.

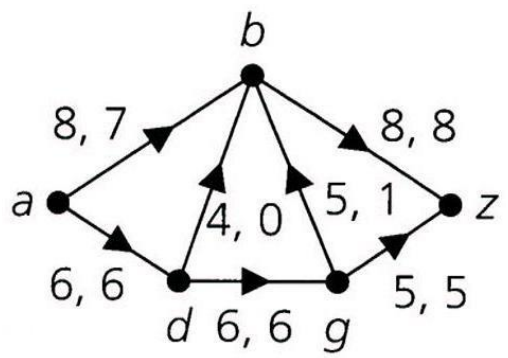
Proof. **Clearly, the capacity constraint holds for each $e \in E$. The conservation constraint also holds for each $v \in V - \{a, z\}$.**



Theorem. F **is maximum iff there is no augmenting** a**-to-**z **path in** N**.**

Proof. **(⇒) trivial.**

(⇐) Find an S **with** $F = c(S)$ **as follows.**



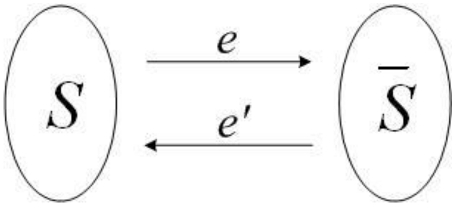
S = $\{v \mid$ **there is an augmenting** a**-to-**v **path in** $N\}$

(= $\{a, b, d, g\}$ ($\overline{S}$ **contains** z)).

$\Rightarrow$ $f(e) = c(e)$ **if** $e \in E(S;\overline{S})$, **and** $f(e) = 0$ **if** $e \in E(\overline{S};S)$

$\Rightarrow$ $F = c(S)$

(Intuitively, F is maximum iff $F = c(S)$ iff $f(e) = c(e)$ and $f(e') = 0$ (see below).

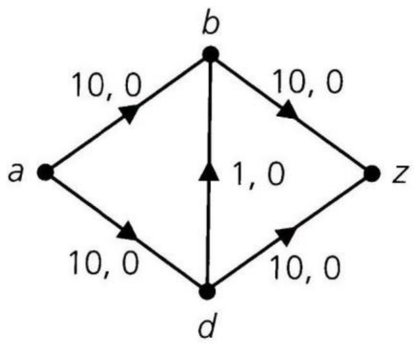


$\Rightarrow$ **there is no augmenting *a*-to-*z* path passing the cross section induced by S and $\overline{S}$)**

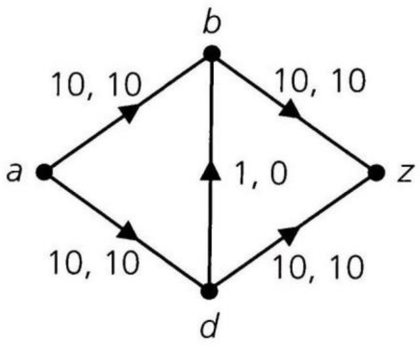
Two flaws of Ford & Fulkerson's algorithm:

(1) The capacities must be rational numbers. If they are irrational numbers, then Ford & Fulkerson's algorithm may cause an infinite sequence of flow augmentations, and the flow finally converges to a value that is one fourth of the maximum total flow.

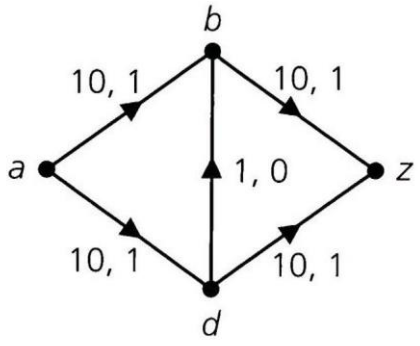
(2) Ford & Fulkerson's algorithm takes exponential time in the worst case.



Select (*a*, *b*, *z*) and (*a*, *d*, *z*) as augmenting paths.



Select (*a*, *d*, *b*, *z*) and (*a*, *b*, *d*, *z*) alternately as augmenting paths.



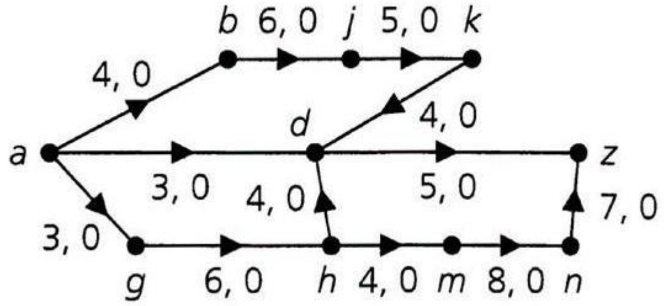
Edmonds & Karp's algorithm can overcome the two flaws of Ford & Fulkerson's algorithm.

- **Use BFS to find shortest augmenting paths iteratively, where unsaturated forward edges and backward edges e with $f(e) > 0$ are considered.**
- **A maximum flow is obtained, when no augmenting path is available.**

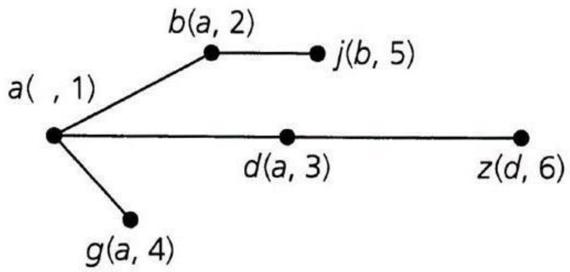
Remarks. Edmonds & Karp's algorithm takes $O(|V|^3|E|)$ time. There are still more efficient algorithms, e.g., by

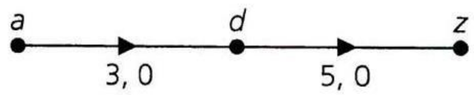
Dinic ($O(|V|^2|E|)$ time);
Karzanov ($O(|V|^3)$ time);
Malhotra, Pramodh Kumar, Maheshwari ($O(|V|^3)$ time);
Cherkassky ($O(|V|^2|E|^{1/2})$ time).

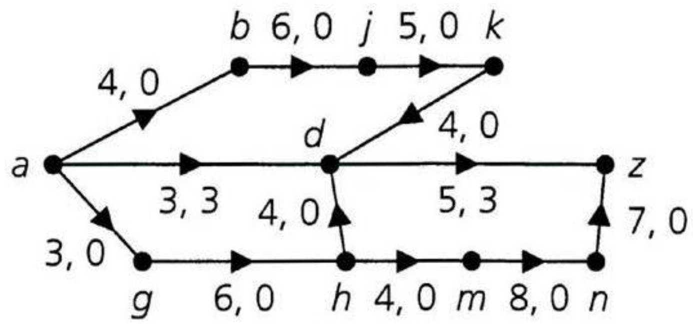
Initially,



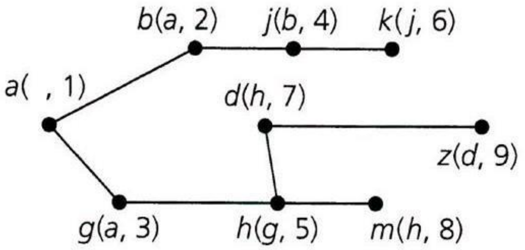
First iteration,

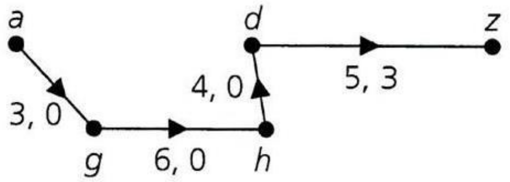


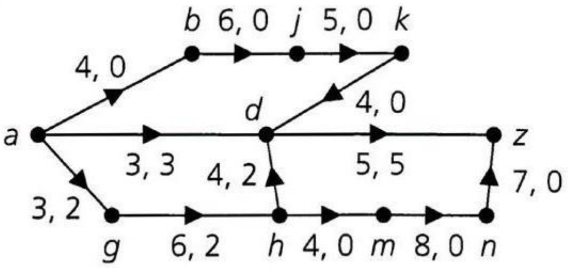




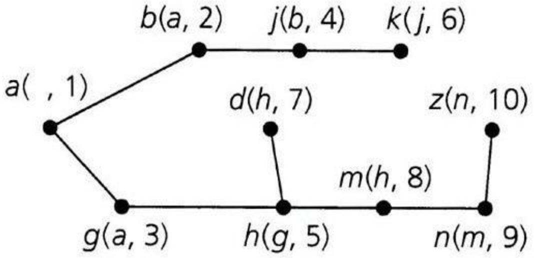
Second iteration,

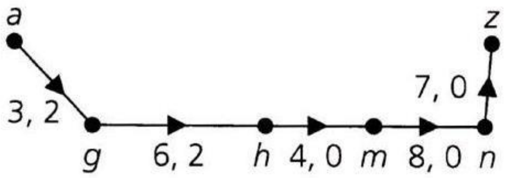


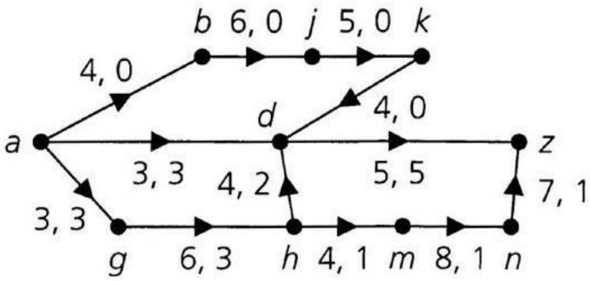




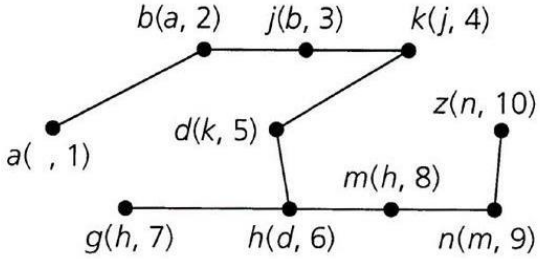
Third iteration,

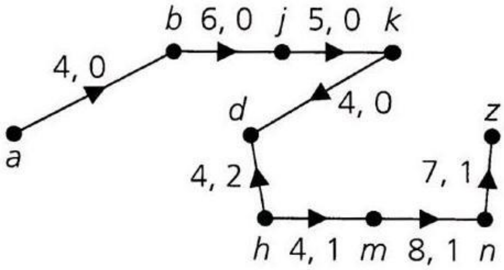


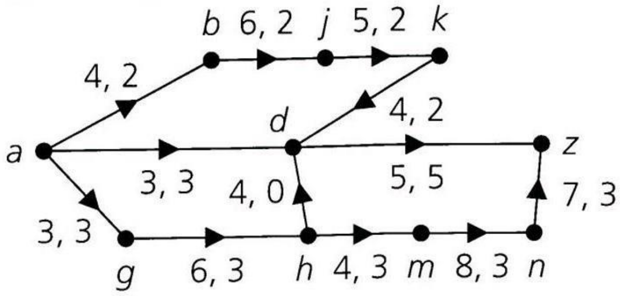




Fourth, iteration,

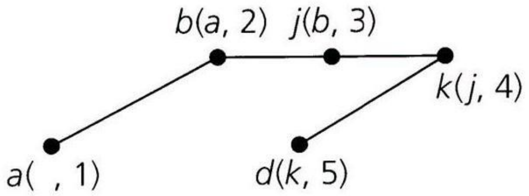


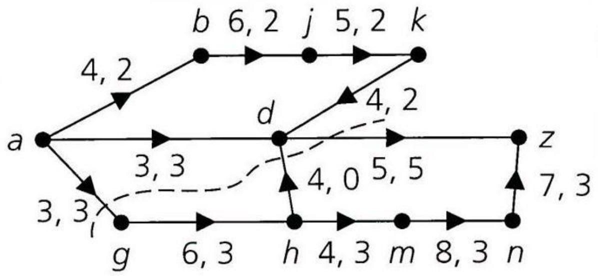




No further augmenting path can be found.

$F^* = 8$ **and** $S^* = \{a, b, d, j, k\}$.





• Coloring, Chromatic Number, and Chromatic Polynomial

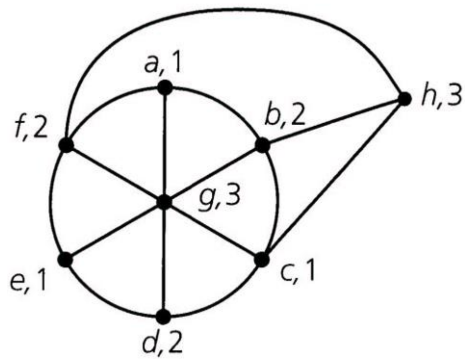
A *proper coloring* of a graph *G* is an assignment of colors to the vertices of *G* so that no two adjacent vertices are assigned with the same color.

The *chromatic number* of *G*, denoted by $\chi(G)$, is the smallest number of colors needed to properly color *G*.

A graph is *k-colorable* iff it can be properly colored with *k* colors.

Ex. $\chi(K_n) = n$.

Ex. **The following graph G is 3-colorable. Since it contains a K_3, it has $\chi(G) \geq 3$. Therefore, $\chi(G) = 3$.**



Theorem. **An undirected graph $G = (V, E)$ is 2-colorable (i.e., bipartite) iff G has no cycle of odd length.**

Define $P(G, \lambda)$ to be the number of different ways to properly color G with at most λ distinct colors.

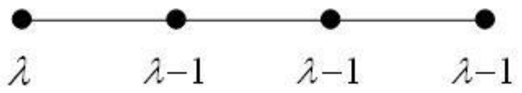
Considering λ a variable, $P(G, \lambda)$ is a polynomial function, called the *chromatic polynomial* of G.

The value of $P(G, \lambda)$ is equal to the number of functions $f: V \to \{1, 2, \ldots, \lambda\}$ so that $f(u) \neq f(v)$ if $(u, v) \in E$.

Ex. $P(K_n, \lambda) = 0$ **as** $\lambda < n$.

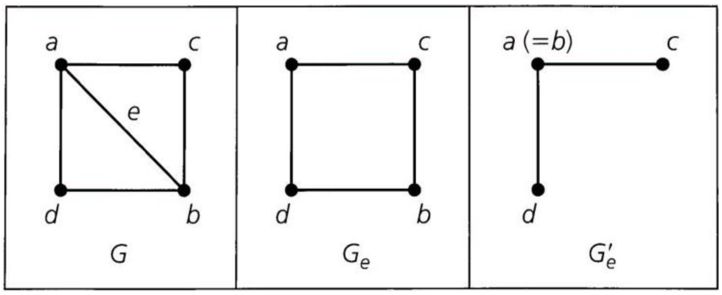
When $\lambda \geq n$, $P(K_n, \lambda) = \lambda(\lambda - 1)(\lambda - 2) \ldots (\lambda - n + 1)$.

Ex. If G **is a path on** n **vertices, then** $P(G, \lambda) = \lambda(\lambda - 1)^{n-1}$.



G_e : **the graph obtained by deleting an edge e of G.**

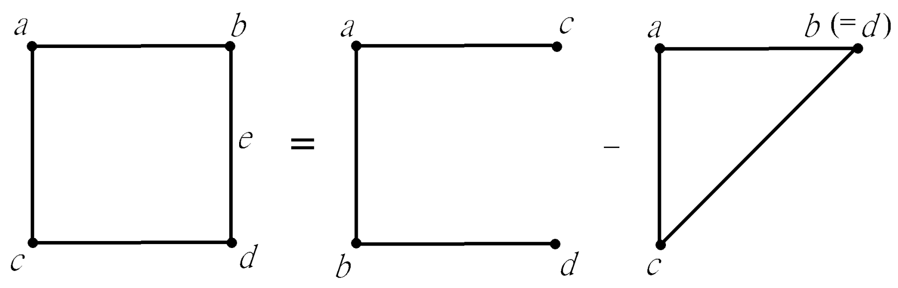
G_e' : **the graph obtained from G by identifying the two end vertices of e.**



Theorem. $P(G_e, \lambda) = P(G, \lambda) + P(G_e', \lambda)$.

(or $P(G, \lambda) = P(G_e, \lambda) - P(G_e', \lambda)$.)

Ex. Compute $P(G, \lambda)$ for G being a cycle of length 4.



$$P(G_e, \lambda) = \lambda(\lambda - 1)^3.$$

$$P(G_e', \lambda) = \lambda(\lambda - 1)(\lambda - 2).$$

$$P(G, \lambda) = \lambda(\lambda - 1)^3 - \lambda(\lambda - 1)(\lambda - 2)$$
$$= \lambda^4 - 4\lambda^3 + 6\lambda^2 - 3\lambda.$$

$$P(G, 1) = 0,\ P(G, 2) = 2 \quad \Rightarrow \quad \chi(G) = 2.$$

The constant term in $P(G, \lambda)$ is 0, for otherwise $P(G, 0) \neq 0$.

When $|E| > 0$, the sum of the coefficients in $P(G, \lambda)$ is 0, for otherwise $P(G, 1) \neq 0$.

(do Exercise #12)