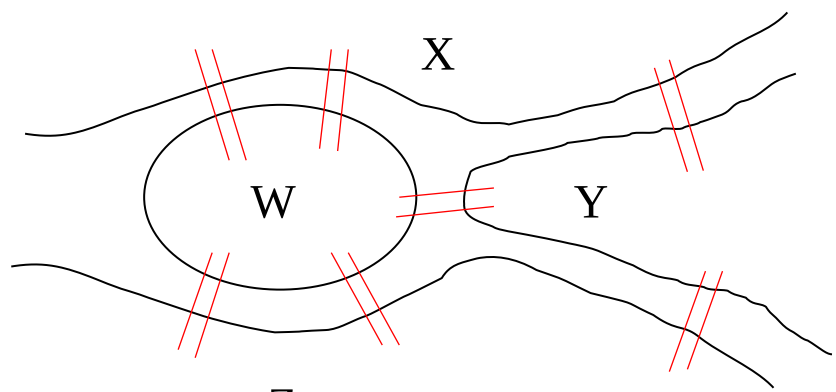
Chapter 1
Fundamental Concept

The Königsberg Bridge Problem

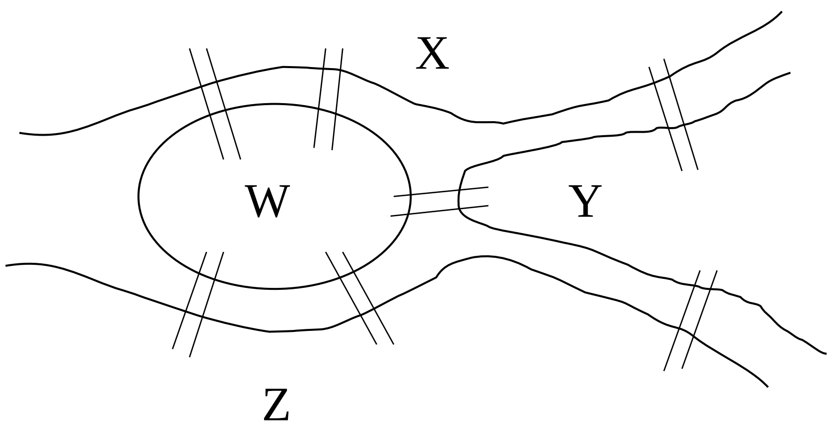
- Königsber is a city on the Pregel river in Prussia
- The city occupied two islands plus areas on both banks
- Problem:

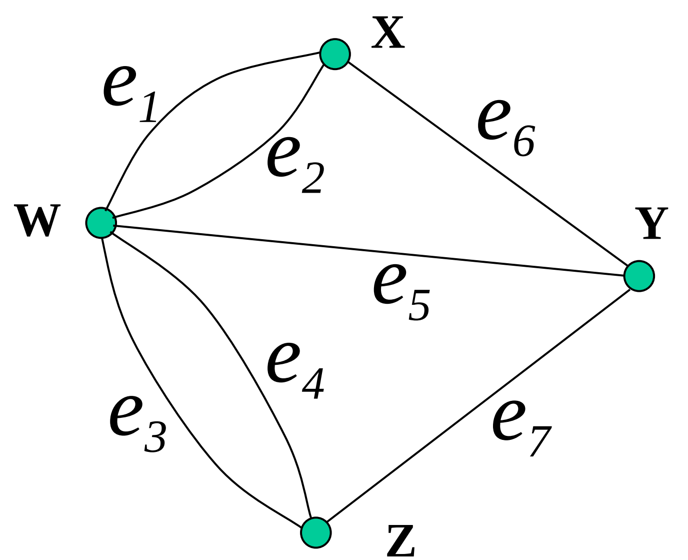
  Whether they could leave home, cross every bridge exactly once, and return home.



A Model

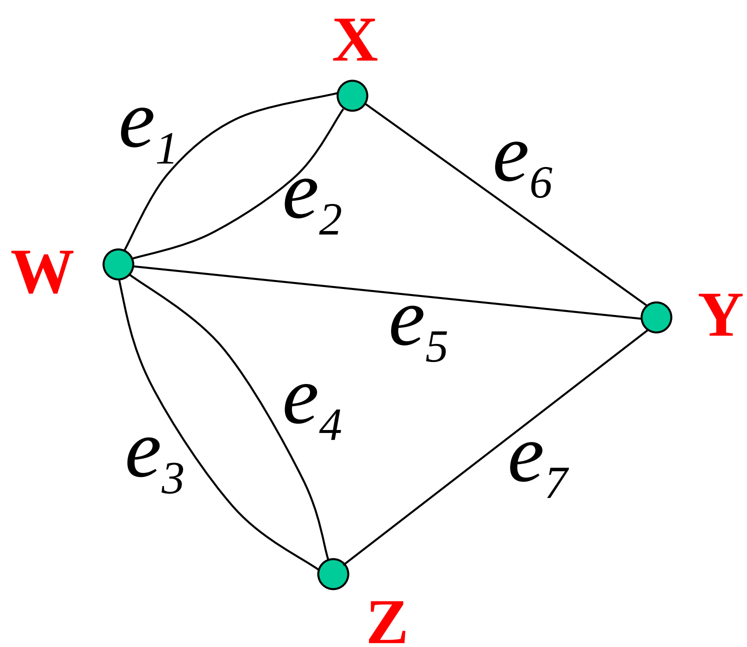
- A ***vertex***: a region
- An ***edge***: a path(bridge) between two regions





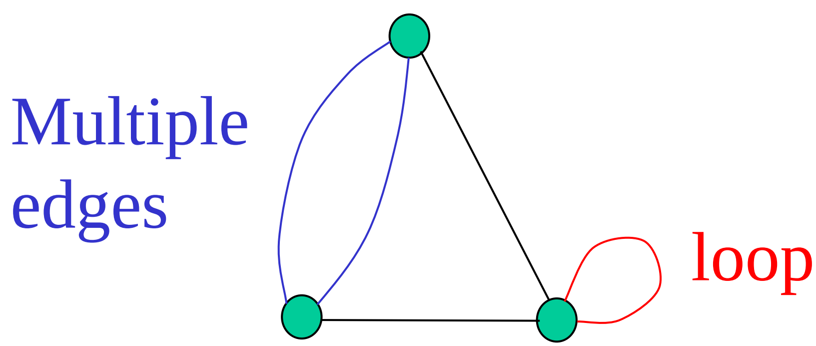
What Is a Graph?

- A graph G is a triple consisting of:
  - A vertex set $V(G)$
  - An edge set $E(G)$
  - A relation between an edge and a pair of vertices



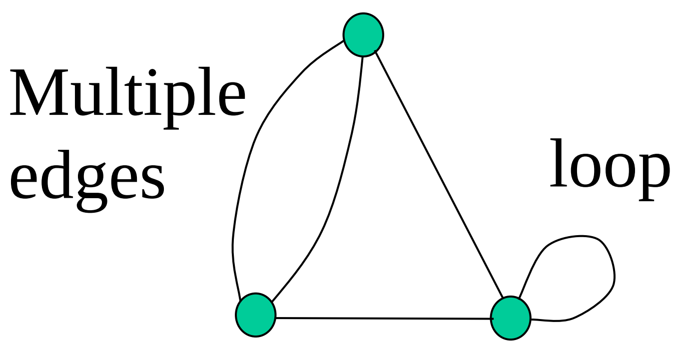
Loop, Multiple edges

- ***Loop***: An edge whose endpoints are equal
- ***Multiple edges***: Edges have the same pair of endpoints

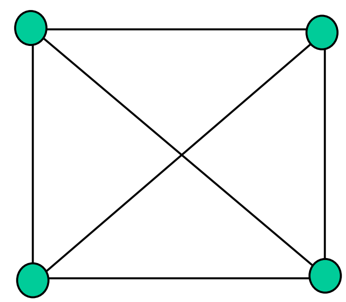


Simple Graph

- ***Simple graph***: A graph has no loops or multiple edges



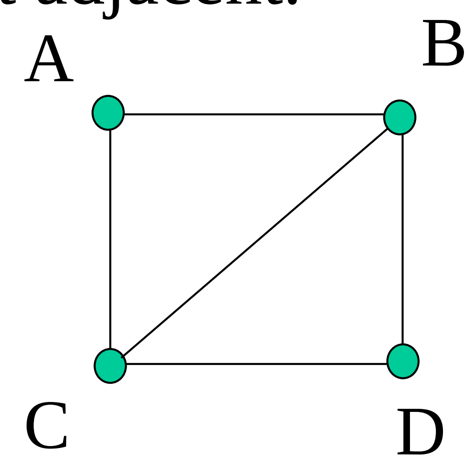
It is not simple.



It is a simple graph.

Adjacent, neighbors

- Two vertices are *adjacent* and are *neighbors* if they are the endpoints of an edge.
- Example:
  - A and B are adjacent.
  - A and D are not adjacent.

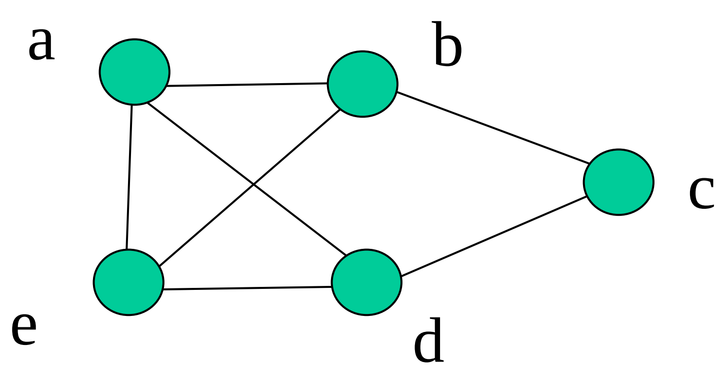


Finite Graph, Null Graph

- ***Finite graph***: an graph whose vertex set and edge set are finite.
- ***Null graph***: the graph whose vertex set and edges are empty.

Path and Cycle

- ***Path***: a sequence of distinct vertices such that two consecutive vertices are adjacent.
  - Example: (a, d, c, b, e) is a path
  - (a, b, e, d, c, b, e, d) is not a path; it is a walk.
- ***Cycle***: a closed Path
  - Example: (a, d, c, b, e, a) is a cycle



Walks, Trails 1.2.2

- A ***walk*:** a list of vertices and edges $v_0, e_1, v_1, \ldots,$ e_k, v_k such that, for $1 \le i \le k$, the edge e_i has endpoints v_{i-1} and v_i.

- A ***trail*** : a walk with no repeated edge.

Paths 1.2.2

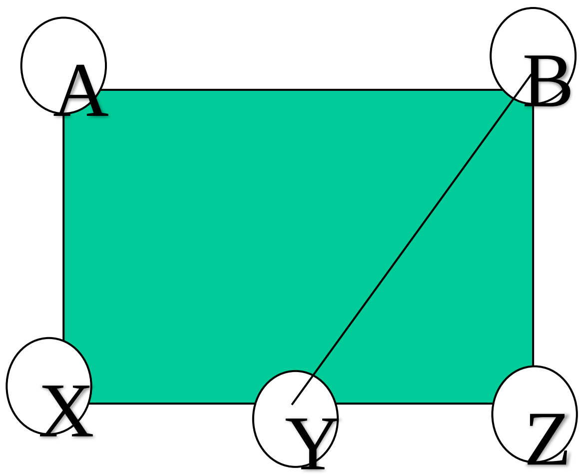
- A u,v-walk or u,v-trail has first vertex u and last vertex v; these are its endpoints.

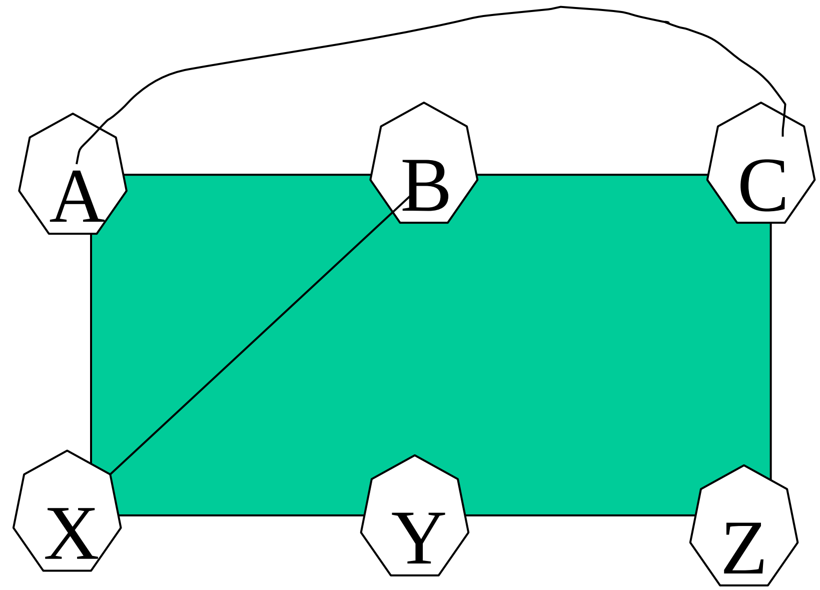
- A ***u,v-path***: a u,v-trail with no repeated vertex.

- The ***length*** of a walk, trail, path, or cycle is its number of edges.

- A walk or trail is ***closed*** if its endpoints are the same.

Let G be the Graph as follows find

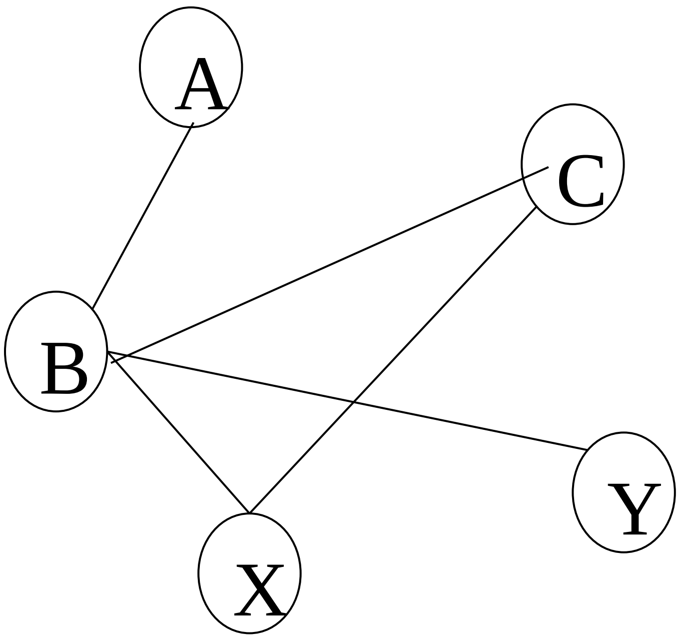




(a) All the simple path from vertex A to Z

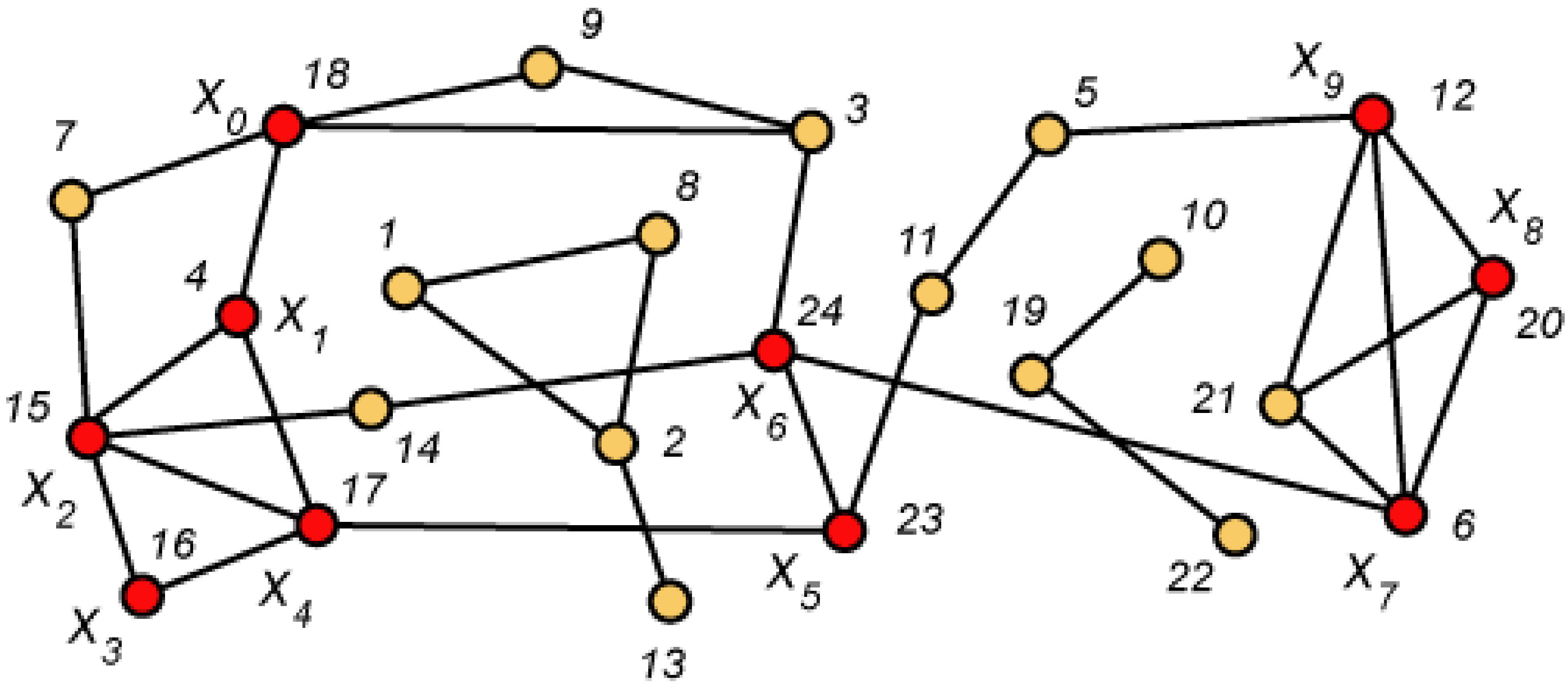
(b) A trail from A to Z, Find cycle C_K, K=3,4,5,6 If possible

Ex. Determine whether each of the following is a path, trail or cycle.



(i) (B,X,C,B)
(ii) (X,A,B,Y)
(iii)(B,X,Y,B)
(iv)(B,A,X,C,B,Y)
(v) (X,C,A,B,Y)
(vi)(X,B,A,X,C)
(vii)(X,B,A,X,B)
(viii)(A,B,C,X,B,A)
(ix)(X,C,B,A)

A Path from 18 to 12



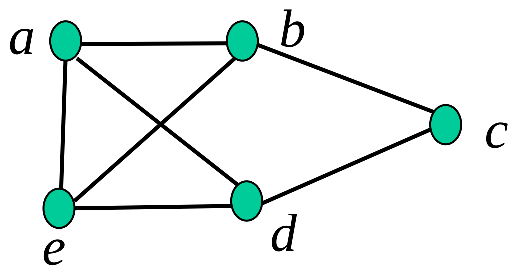
Subgraphs

- A ***subgraph*** of a graph G is a graph H such that:
  - $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ and
  - The assignment of endpoints to edges in H is the same as in G.

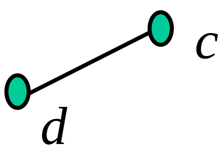
Subgraphs

- Example: H_1, H_2, and H_3 are subgraphs of G

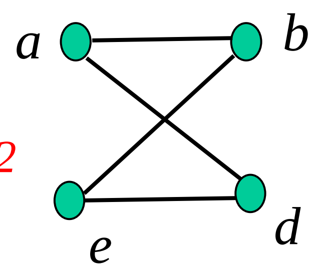
G



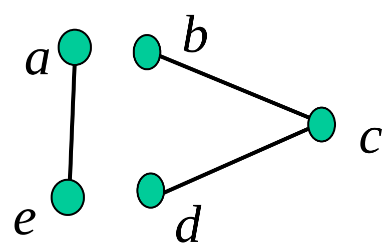
H_1

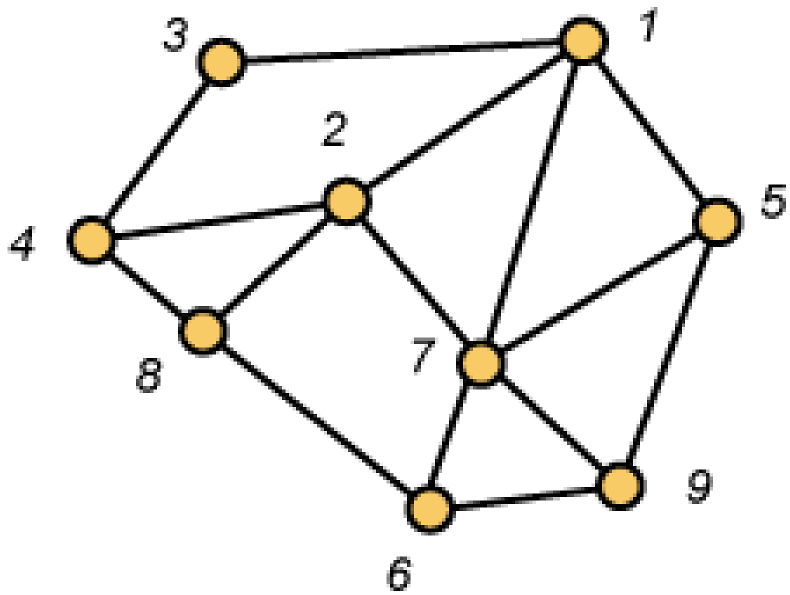


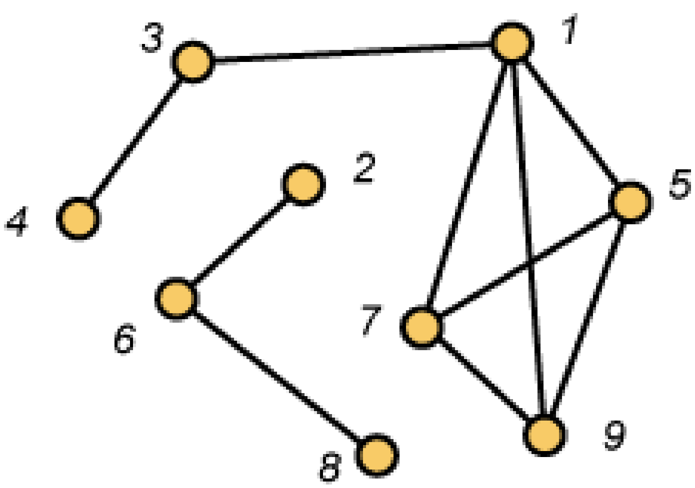
H_2

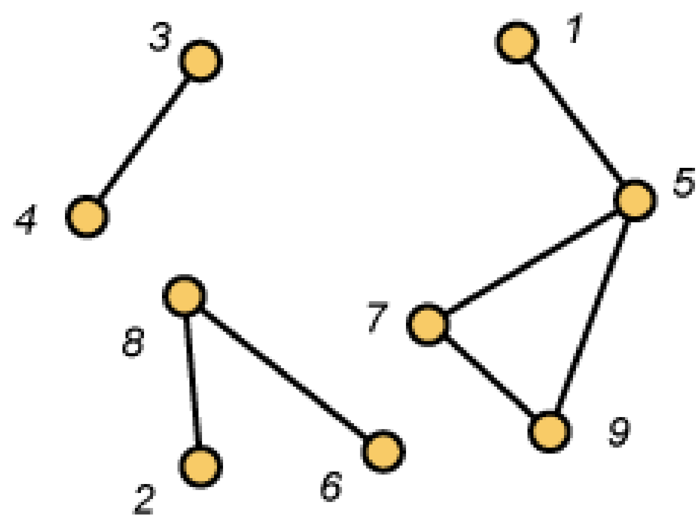


H_3







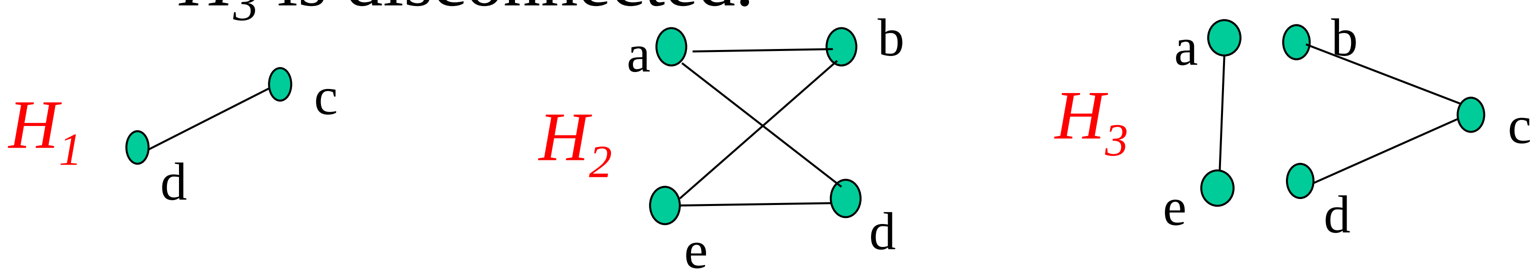


NO

YES

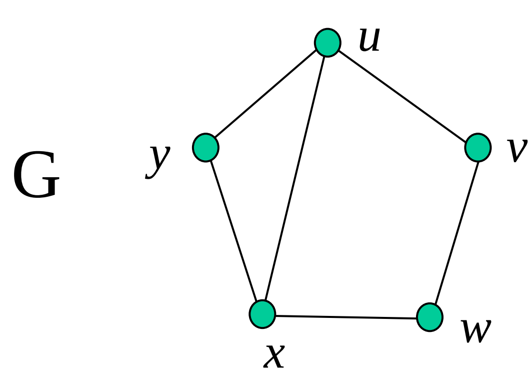
Connected and Disconnected

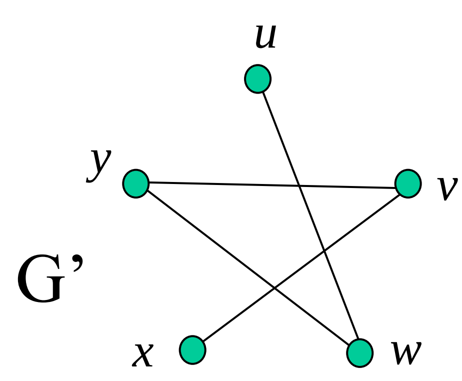
- ***Connected***: There exists at least one path between two vertices.
- ***Disconnected***: Otherwise
- Example:
  - H_1 and H_2 are connected.
  - H_3 is disconnected.



Complement

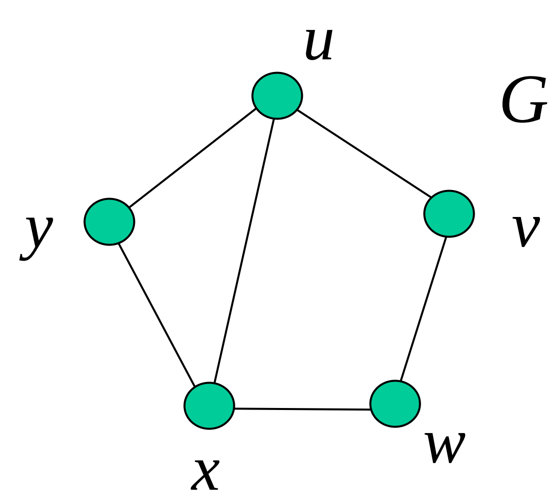
- ***Complement of G***: The complement G' of a simple graph G :
  - A simple graph
  - $V(G') = V(G)$
  - $E(G') = \{ uv \mid uv \notin E(G) \}$





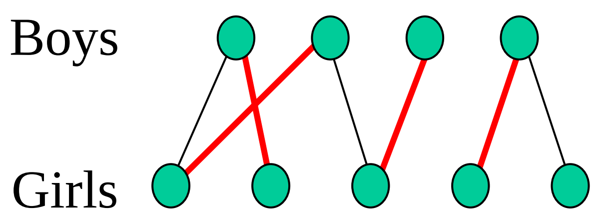
Clique and Independent set

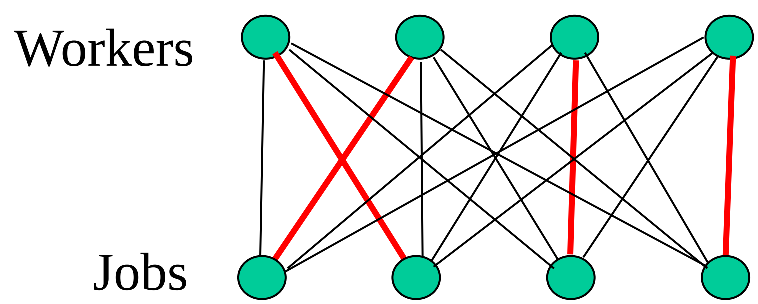
- A ***Clique*** in a graph: a set of pairwise adjacent vertices (a complete subgraph)
- An ***independent set*** in a graph: a set of pairwise nonadjacent vertices.
- Example:
  - $\{x, y, u\}$ is a clique in G.
  - $\{u, w\}$ is an independent set.



Bipartite Graphs

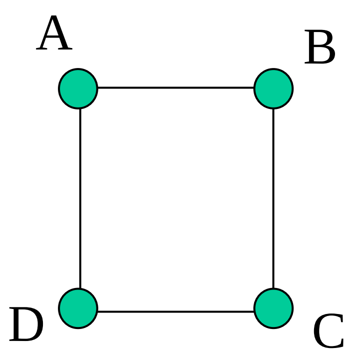
- A graph G is ***bipartite*** if $V(G)$ is the union of two disjoint independent sets called ***partite sets of G***
- *Also*: The vertices can be partitioned into two sets such that each set is independent
- Matching Problem
- Job Assignment Problem

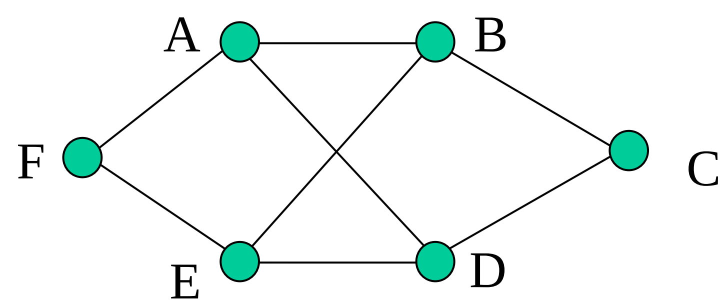


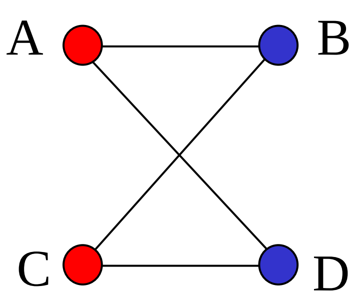


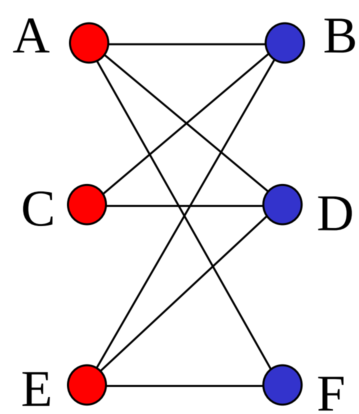
Theorem: A graph is bipartite if and only if it has no odd cycle.

- Examples:





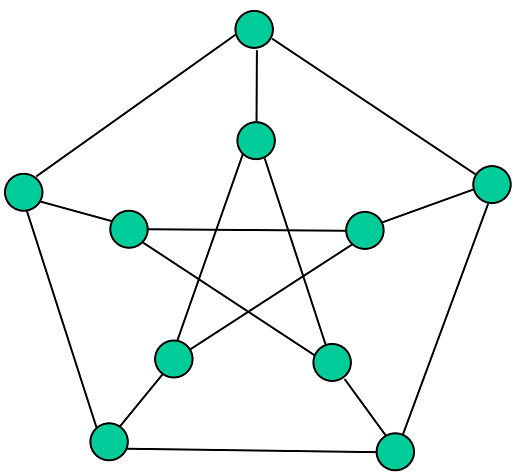


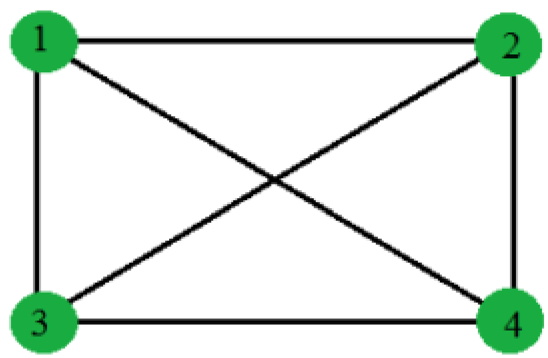


Regular

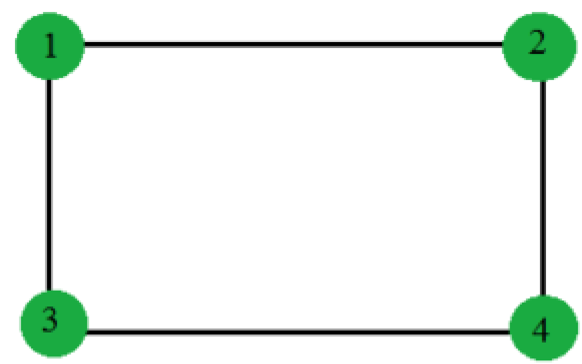
A graph is called regular graph if degree of each vertex is equal. A graph is called K regular if degree of each vertex in the graph is K .

3-regular



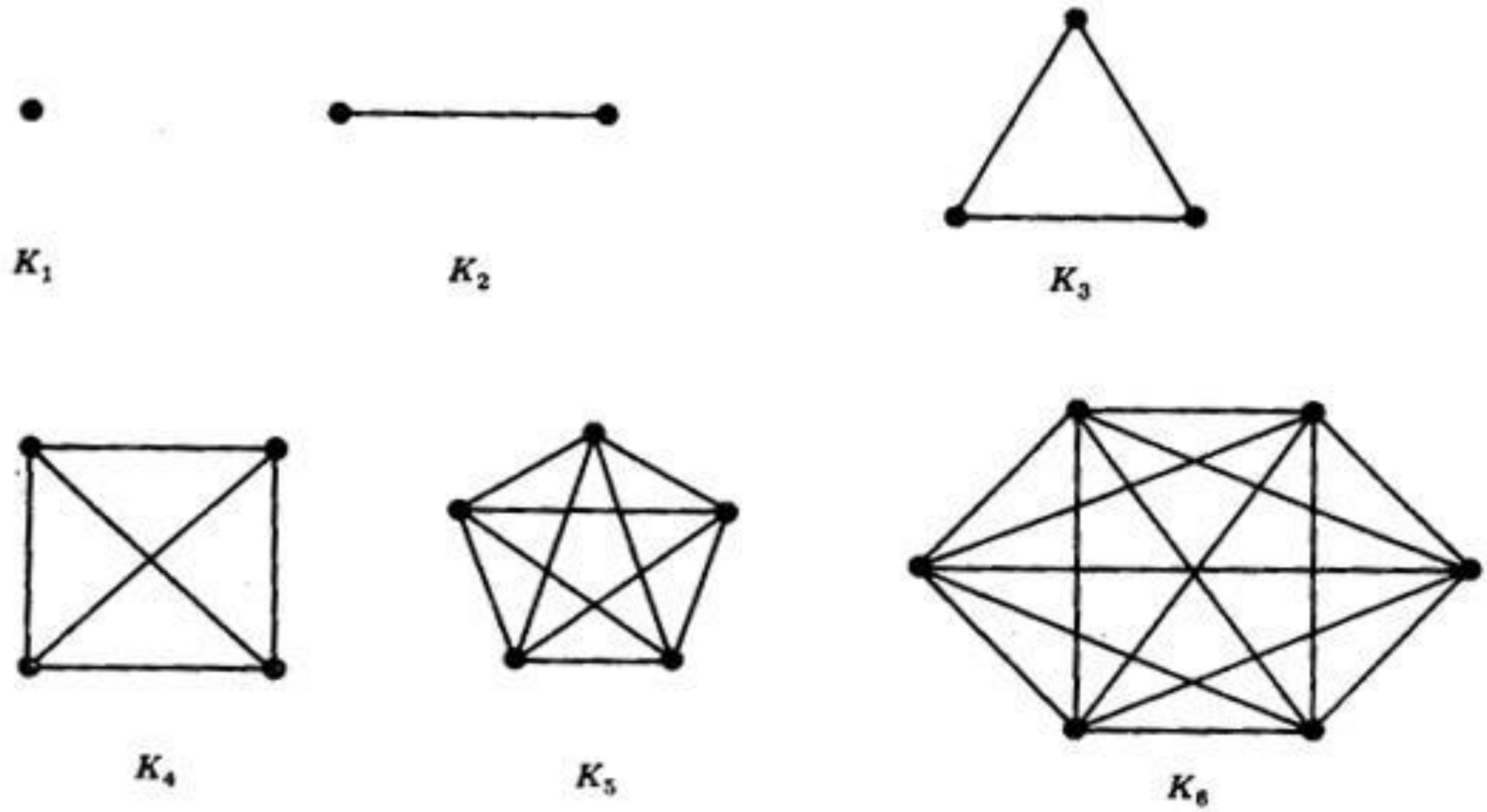


3 Regular



2 Regular

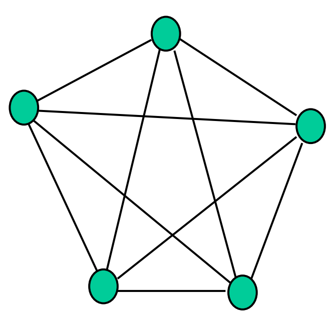
Complete Graph



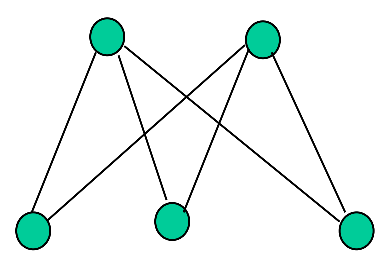
Complete Graph: A complete graph is a graph in which each vertex is connected to every other vertex.

Complete Bipartite Graph or Biclique

- ***Complete bipartite graph*** (biclique) is a simple bipartite graph such that two vertices are adjacent if and only if they are in different partite sets.



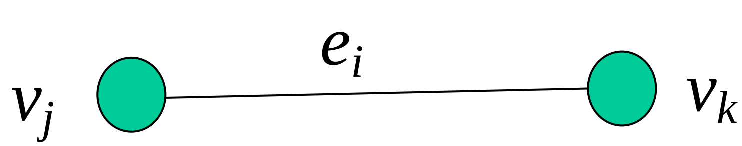
Complete Graph



Complete Bipartite Graph

Adjacency, Incidence, and Degree

- Assume e_i is an edge whose endpoints are (v_j, v_k)
- The vertices v_j and v_k are said to be ***adjacent***.
- The edge e_i is said to be ***incident upon*** v_j
- ***Degree*** of a vertex v_k is the number of edges incident upon v_k. It is denoted as $d(v_k)$



Adjacency matrix

- There is an N x N matrix, where |V| = N , the Adjacenct Matrix (NxN) A = $[a_{ij}]$
- **For undirected graph**

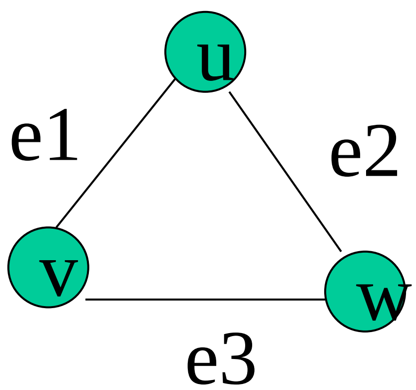
$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G \\ 0 & \text{otherwise} \end{cases}$$

- **For directed graph**

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an edge of } G \\ 0 & \text{otherwise} \end{cases}$$

Adjacency matrix

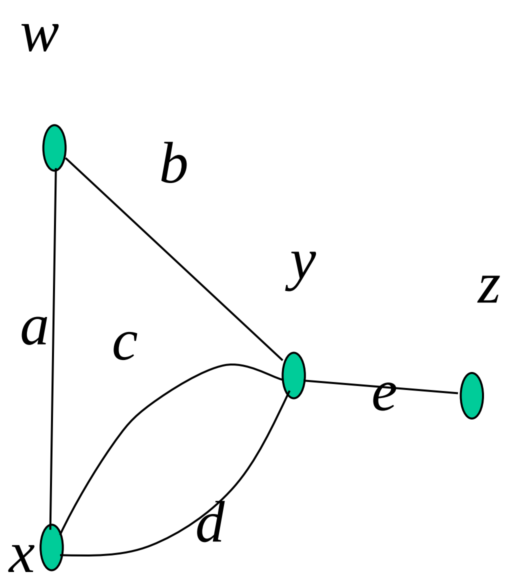
Example: Undirected Graph G (V, E)



	v	u	w
v	0	1	1
u	1	0	1
w	1	1	0

Adjacency matrix

Example: Undirected Graph G (V, E)



$$\begin{array}{c} \\ w \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{cccc} w & x & y & z \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Incidence Matrix

- G = (V, E) be an undirected graph. Suppose that v_1, v_2, v_3, …, v_n are the vertices and e_1, e_2, …, e_m are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the nx m matrix M = [m_{ij}], where

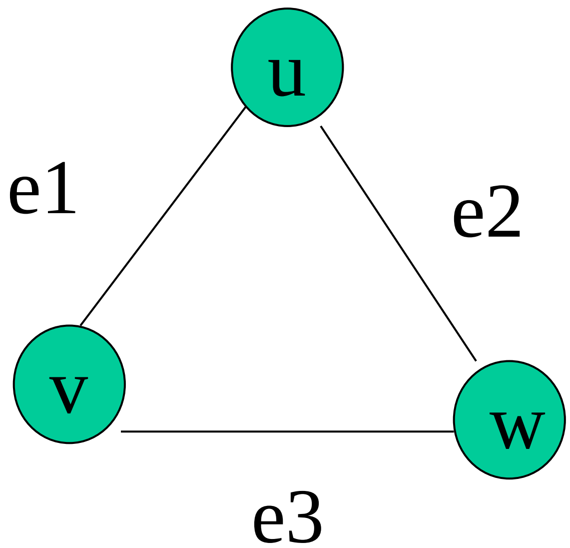
$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i \\ 0 & \text{otherwise} \end{cases}$$

Can also be used to represent :

Multiple edges: by using columns with identical entries, since these edges are incident with the same pair of vertices

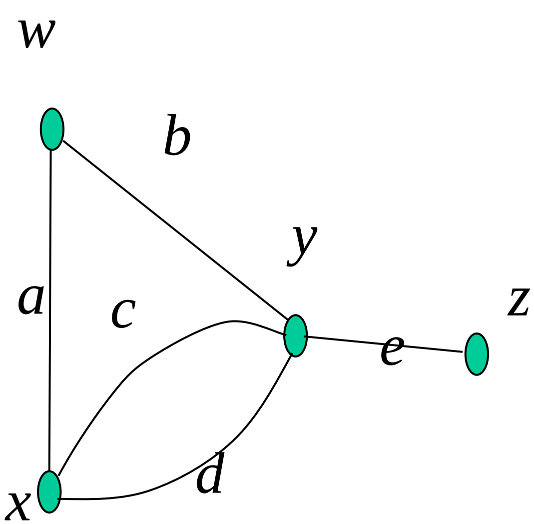
Loops: by using a column with exactly one entry equal to 1, corresponding to the vertex that is incident with the loop

Incidence Matrix



	e_1	e_2	e_3
v	1	0	1
u	1	1	0
w	0	1	1

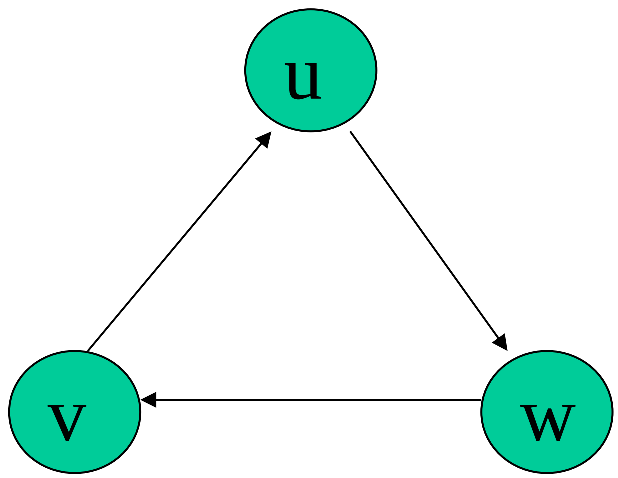
Incidence Matrix



$$\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} w \\ x \\ y \\ z \end{array} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

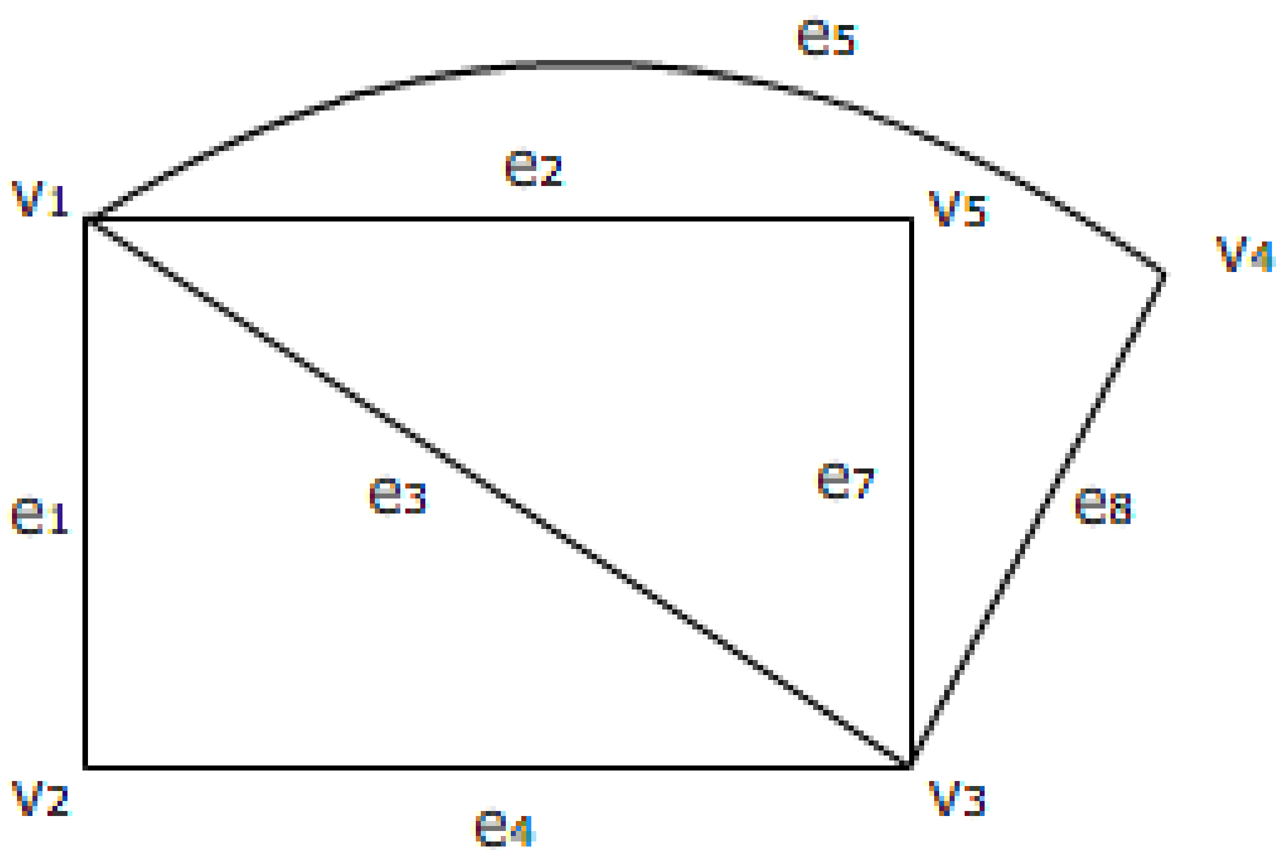
Adjacency matrix

- Example: directed Graph G (V, E)

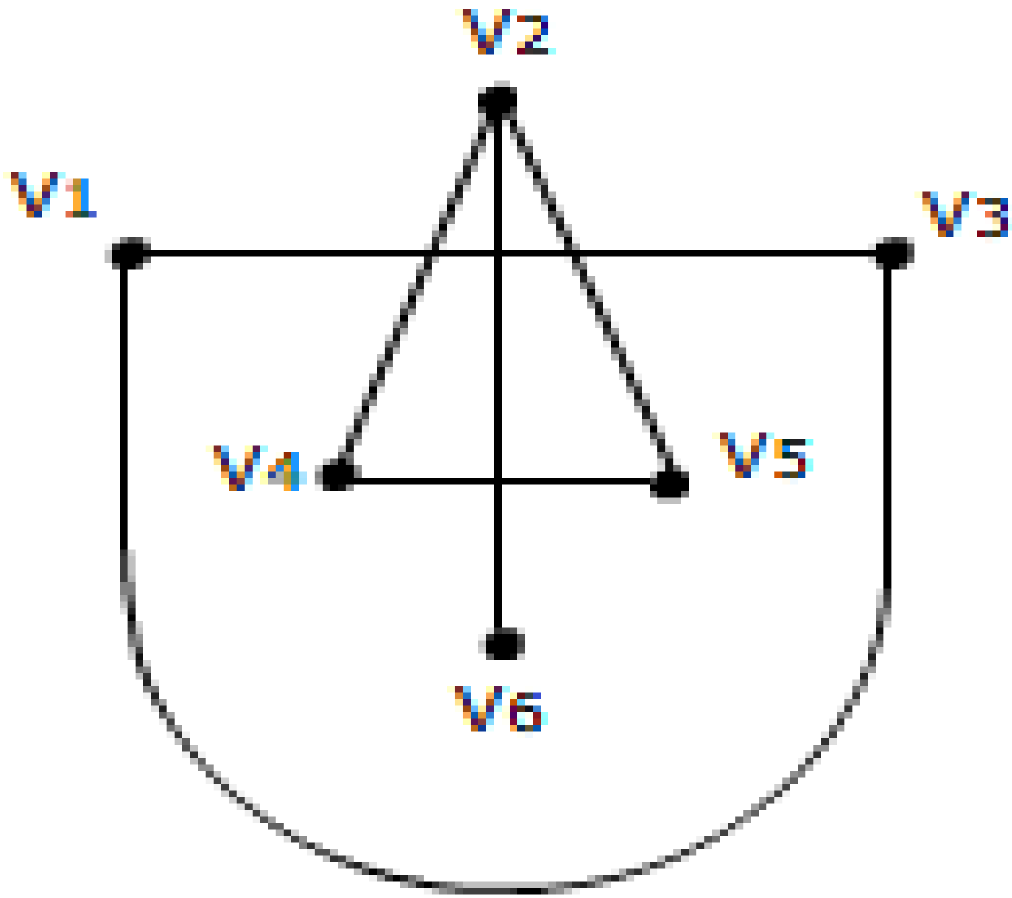


	v	u	w
v	0	1	0
u	0	0	1
w	1	0	0

Adjacency matrix



Adjacency matrix

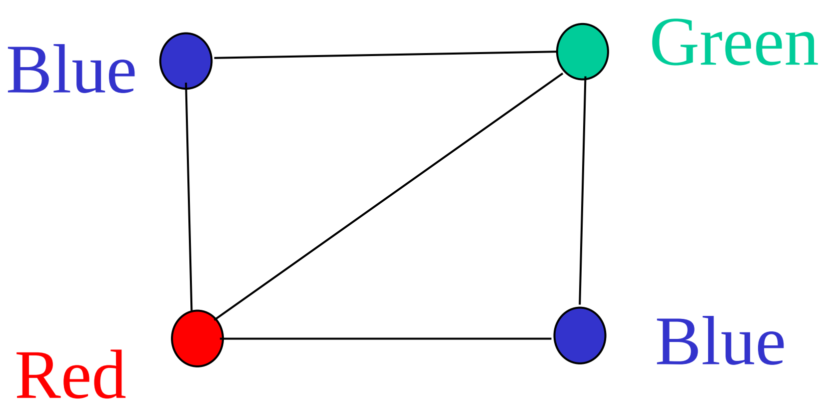


Draw the multigraph whose adjacency matrix is as follows:

$$A = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 0 \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 & 2 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 0 & 3 & 1 & 0 \end{pmatrix}$$

Chromatic Number

- The ***chromatic number*** of a graph G, written $x(G)$, is the minimum number of colors needed to label the vertices so that adjacent vertices receive different colors



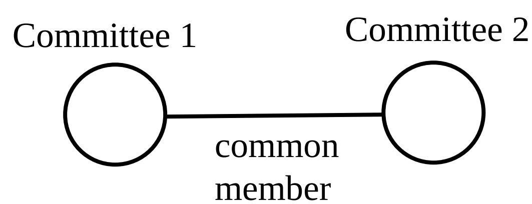
$x(G) = 3$

Maps and coloring

- A ***map*** is a partition of the plane into connected regions
- Can we color the regions of every map using at most four colors so that neighboring regions have different colors?
- Map Coloring $\rightarrow$ graph coloring
  - A region $\rightarrow$ A vertex
  - Adjacency $\rightarrow$ An edge

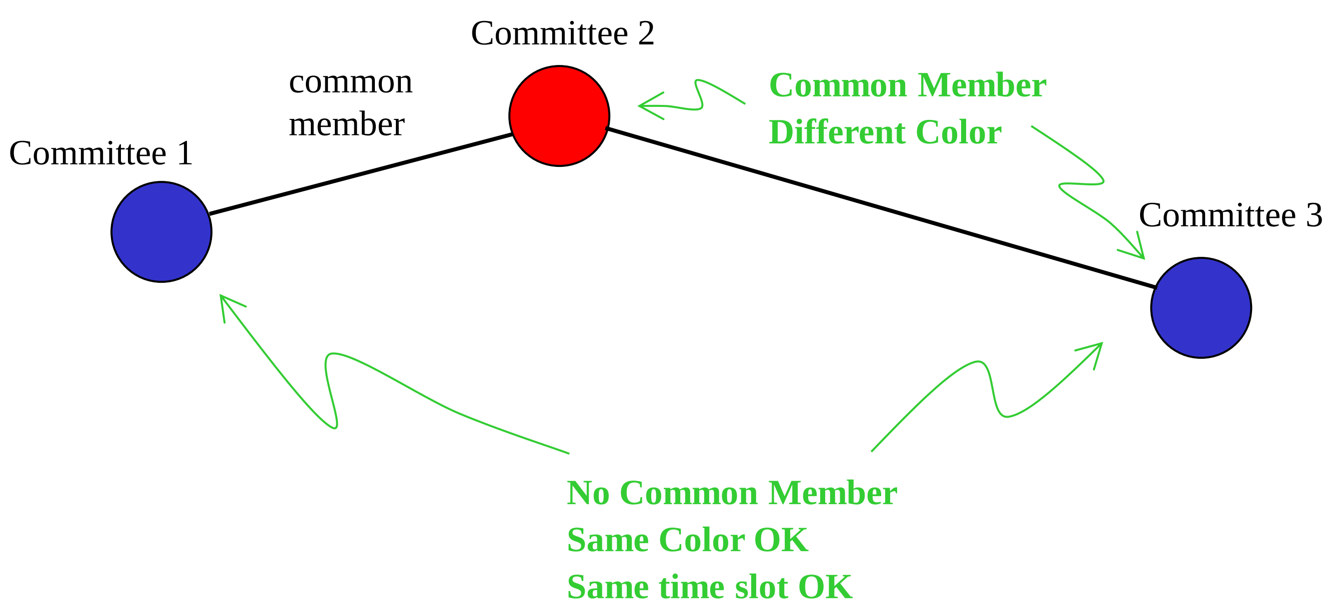
Scheduling and graph Coloring 1

- Model:
  - One committee being represented by a vertex
  - An edge between two vertices if two corresponding committees have common member
  - Two adjacent vertices can not receive the same color



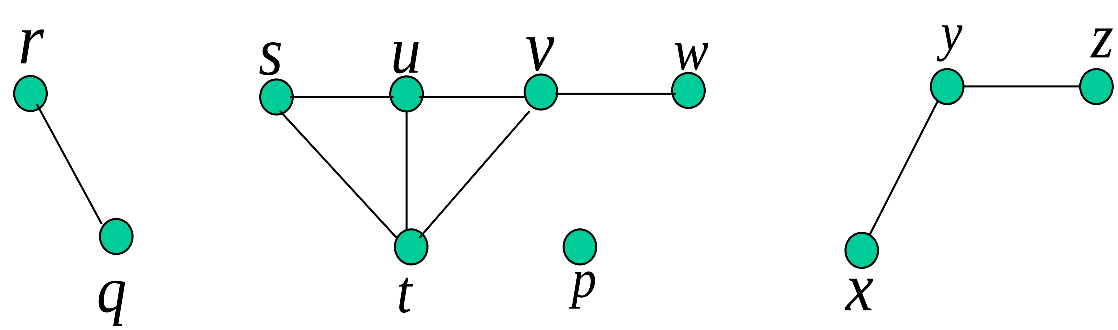
Scheduling and graph Coloring 2

- Scheduling problem is equivalent to graph coloring problem.



Components 1.2.8

- The ***components*** of a graph G are its maximal connected subgraphs.
- A component (or graph) is ***trivial*** if it has no edges; otherwise it is nontrivial.
- An ***isolated vertex*** is a vertex of degree 0.



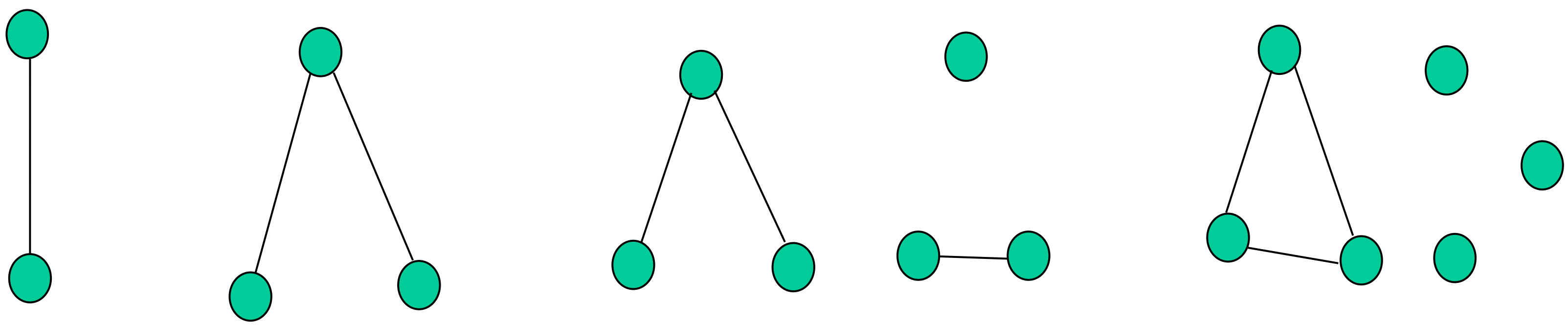
Theorem: Every graph with n vertices and k edges has at least n-k components 1.2.11

Proof:

- An n-vertex graph with no edges has n components
- Each edge added reduces this by at most 1
- If k edges are added, then the number of components is at least n-k

Theorem: Every graph with *n* vertices and *k* edges has at least *n*-*k* components 1.2.11

- Examples:



n=2, *k*=1, 1 component

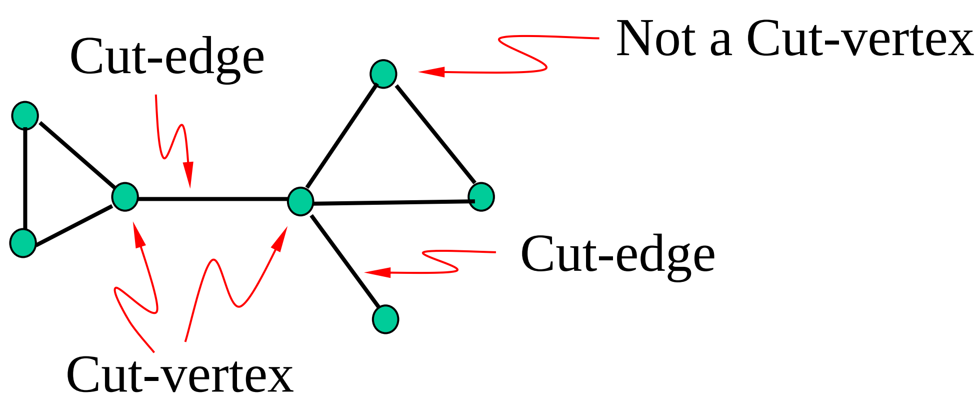
n=3, k=2, 1 component

n=6, k=3, 3 components

n=6, *k*=3, 4 components

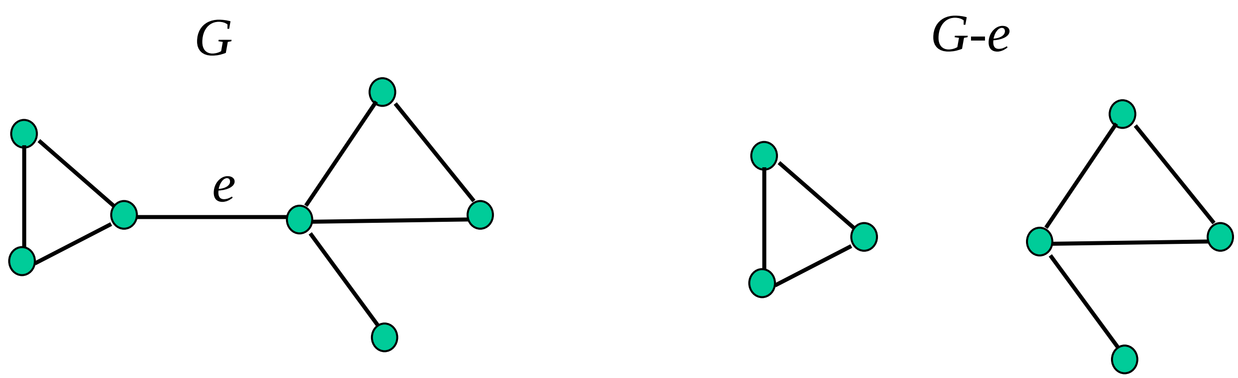
Cut-edge, Cut-vertex 1.2.12

- A ***cut-edge*** or ***cut-vertex*** of a graph is an edge or vertex whose deletion increases the number of components.



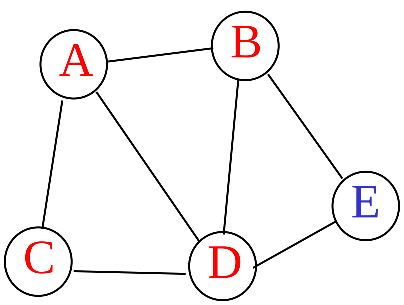
Cut-edge, Cut-vertex 1.2.12

- *G*-*e* or *G*-*M* : The subgraph obtained by deleting an edge *e* or set of edges *M*.
- *G*-*v* or *G*-*S* : The subgraph obtained by deleting a vertex *v* or set of vertices *S*.

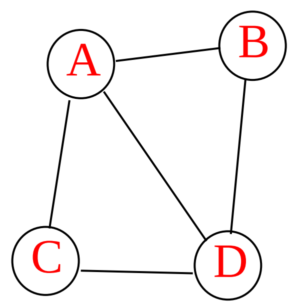


Induced subgraph 1.2.12

- An ***induced subgraph***:
  - A subgraph obtained by deleting a set of vertices.
  - We write $G[T]$ for $G- T'$, where $T' =V(G)-T$;
  - $G[T]$ is the subgraph of G induced by T.
- Example:
  - Assume T:{A, B, C, D}



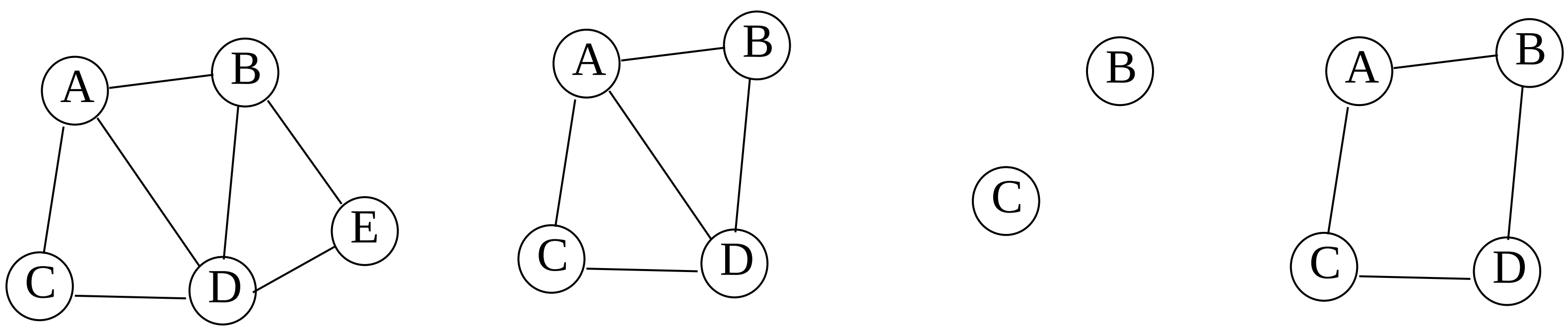
G



$G[T]$

Induced subgraph 1.2.12

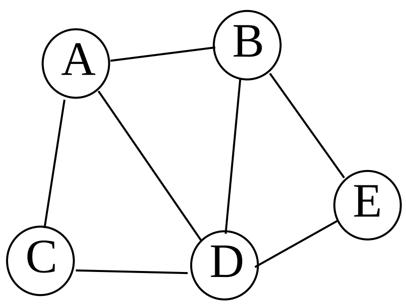
- More Examples:
  - G_2 is the subgraph of G_1 induced by (A, B, C, D)
  - G_3 is the subgraph of G_1 induced by (B, C)
  - G_4 is not the subgraph induced by (A, B, C, D)



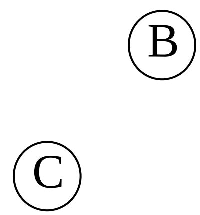
G_1 G_2 G3 G_4

Induced subgraph 1.2.12

- A set S of vertices is an independent set if and only if the subgraph induced by it has no edges.
  - G_3 is an example.



G_1



G3

Eulerian Circuits 1.2.24

- A graph is ***Eulerian*** if it has a closed trail containing all edges.
- We call a closed trail a ***circuit*** when we do not specify the first vertex but keep the list in cyclic order.
- An ***Eulerian circuit*** or ***Eulerian trail*** in a graph is a circuit or trail containing all the edges.

Even Graph, Even Vertex, and Maximal Path 1.2.24

- An ***even graph*** is a graph with vertex degrees all even.
- A vertex is ***odd*** [*even*] when its degree is odd [even].

Theorem: A graph G is Eulerian if and only if it has at most one nontrivial component and its vertices all have even degree. 1.2.26

Adjacency Matrix and Incidence Matrix of a Digraph 1.4.10

- In the ***adjacency matrix*** $A(G)$ of a digraph G, the entry in position i, j is the number of edges from v_i to v_j.
- In the ***incidence matrix*** $M(G)$ of a loopless digraph G, we set $m_{i,j}$=+1 if v_i is the tail of e_j and $m_{i,j}$= -1 if v_i is the head of e_j.

Connected Digraph 1.4.12

- To define connected digraphs, two options come to mind. We could require only that the underlying graph be connected.
- However, this does not capture the most useful sense of connection for digraphs.

Weakly and strongly connected digraphs 1.4.12

- A graph is ***weakly connected*** if its underlying graph is connected.
- A digraph is ***strongly connected*** or ***strong*** if for each *ordered pair u,v* of vertices, there is a path from *u* to *v*.