Priority Queues

Data structure that allows at least the following two operations:

1. Insert
2. DeleteMin

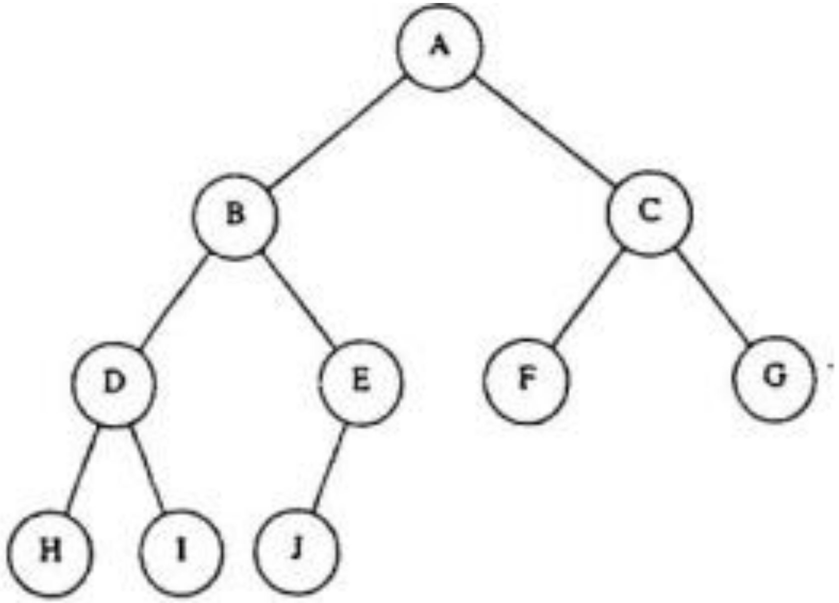
Priority Queues: Simple Implementation

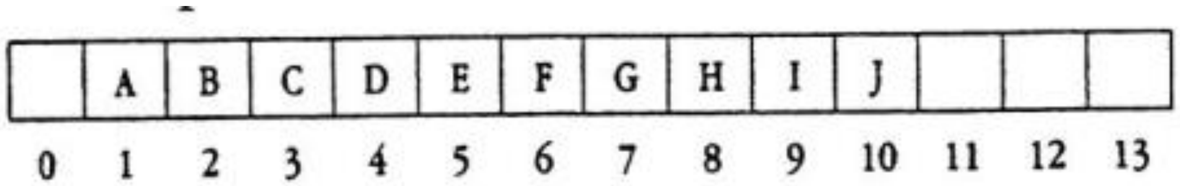
Binary Heap :

Structure Property >> A complete binary tree

It is easy to show that a complete binary heap of height h has between 2^h and $2^{h+1} - 1$ nodes.

Priority Queues: Simple Implementation

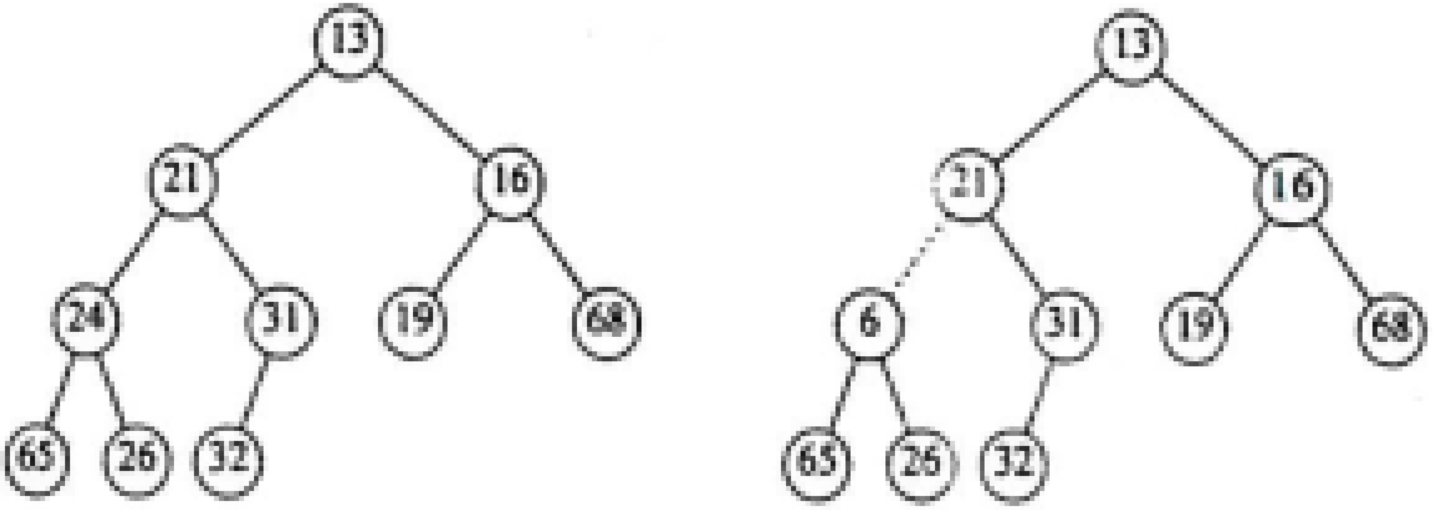




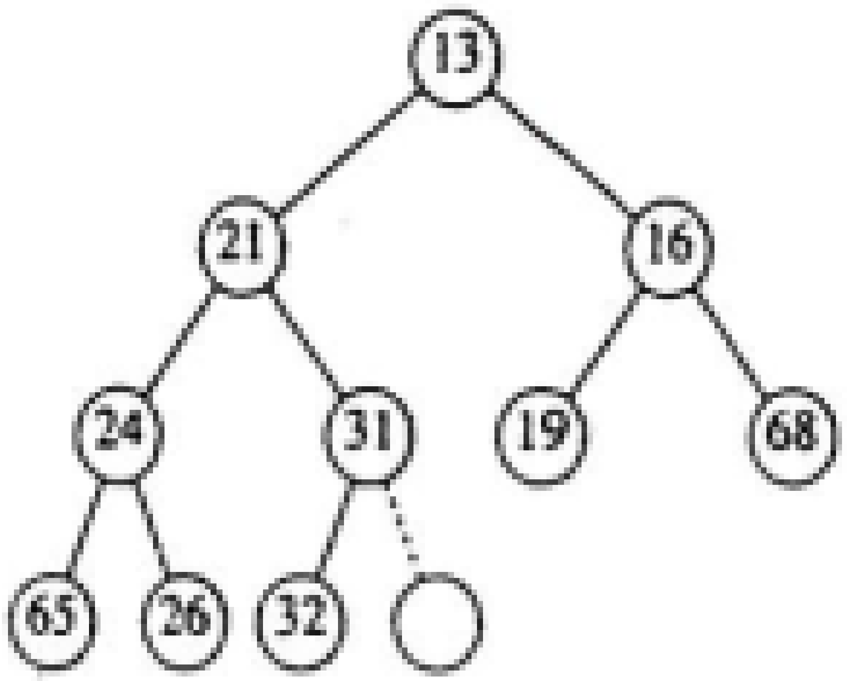
0	1	2	3	4	5	6	7	8	9	10	11	12	13
	A	B	C	D	E	F	G	H	I	J			

Priority Queues: Heap order property

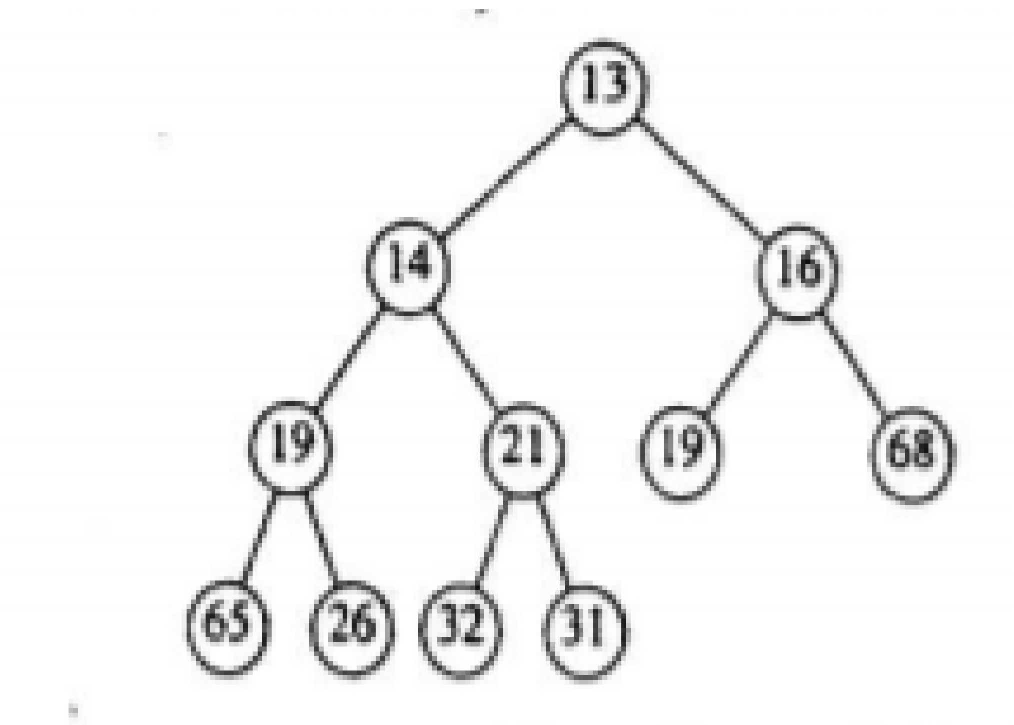
The smallest element should be at the root



Priority Queues: Insert

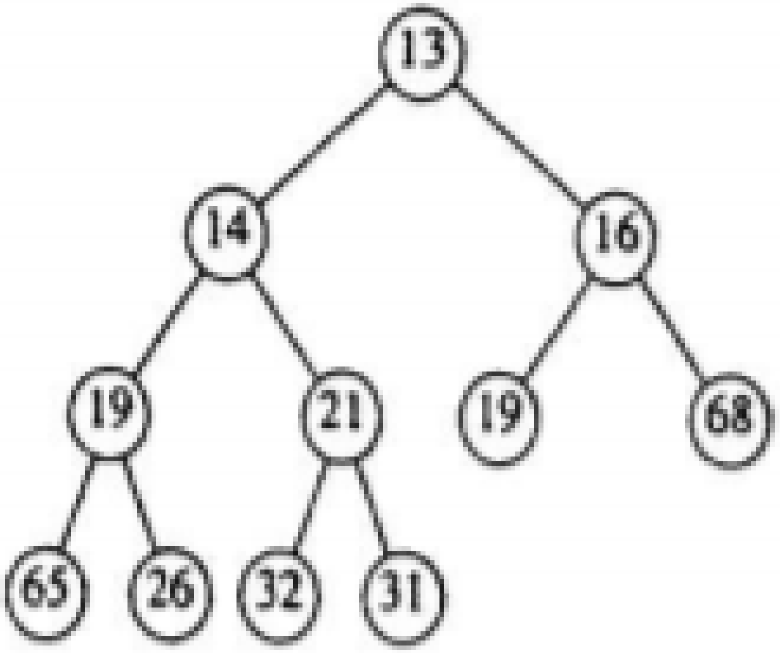


Priority Queues: DeleteMin



Percolate down

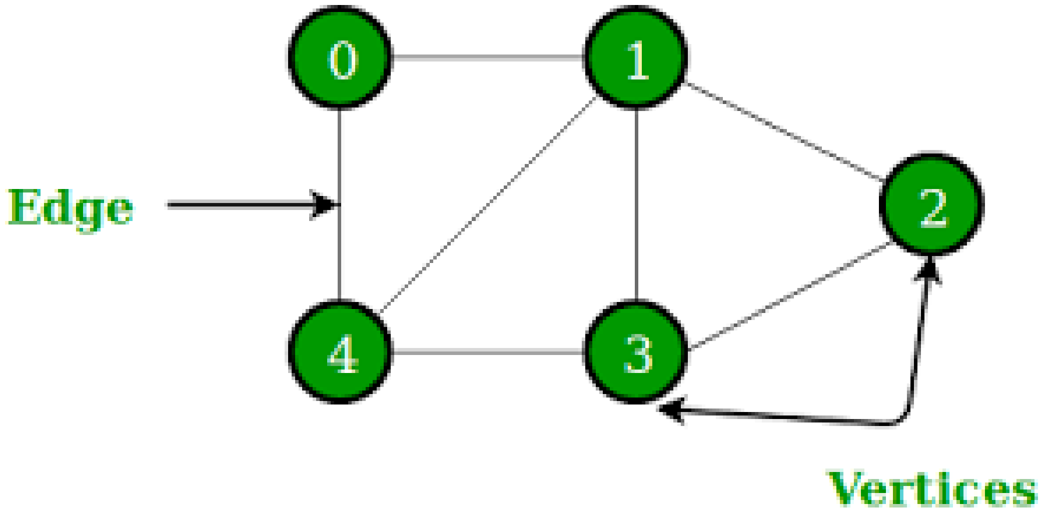
Priority Queues: Build heap



Graph Algorithms

Graph

Graph G=(V,E) consists of a set of vertices or node(V) and a set of edges(E)



Edges

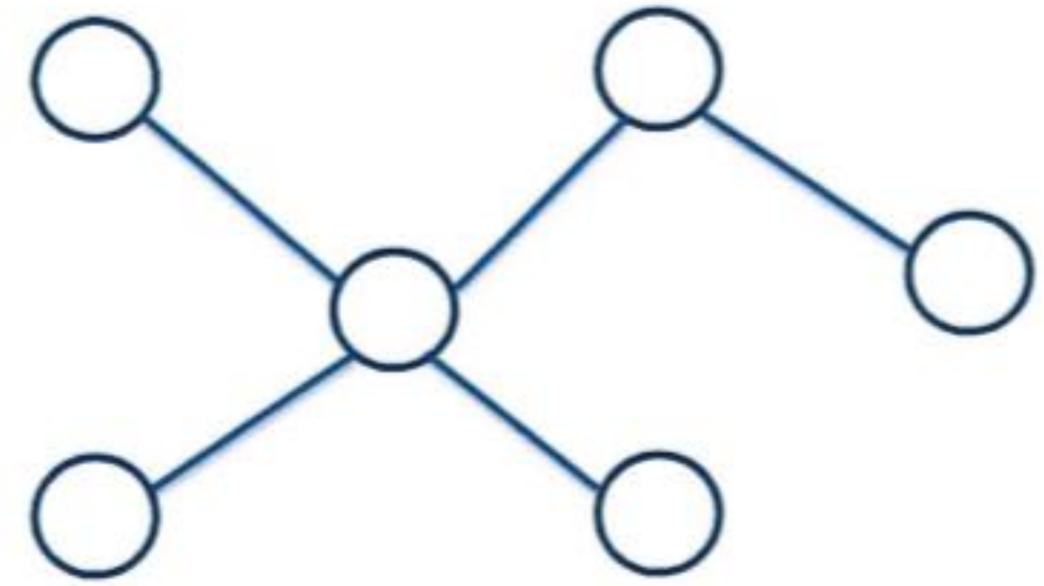
Edges is a pair (v,w)

-Undirected

-Directed

-Weight

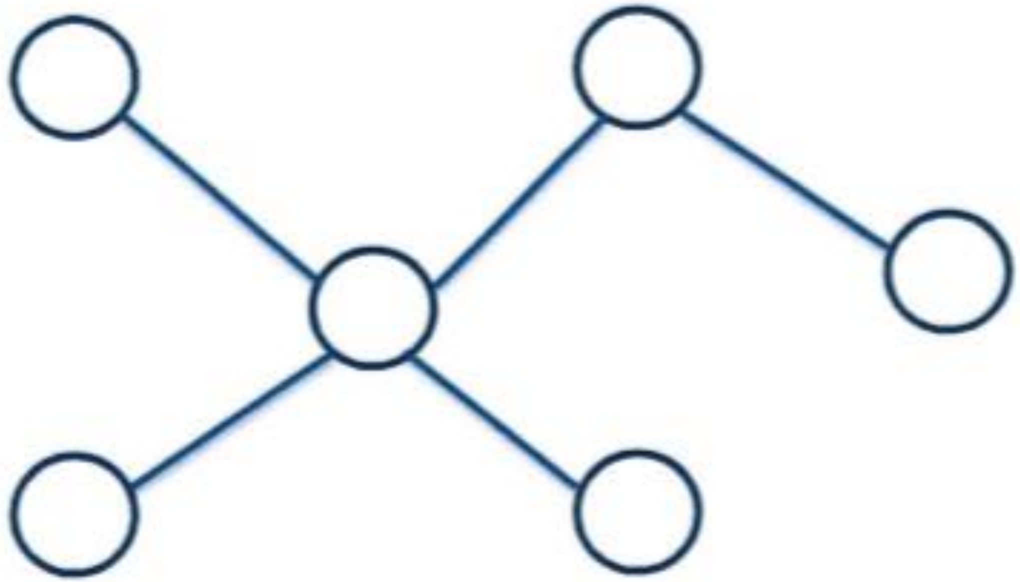
-Unweight



Path

-Unweight

-weighted



Cycle

A cycle in a directed graph is a path of length at lest 1

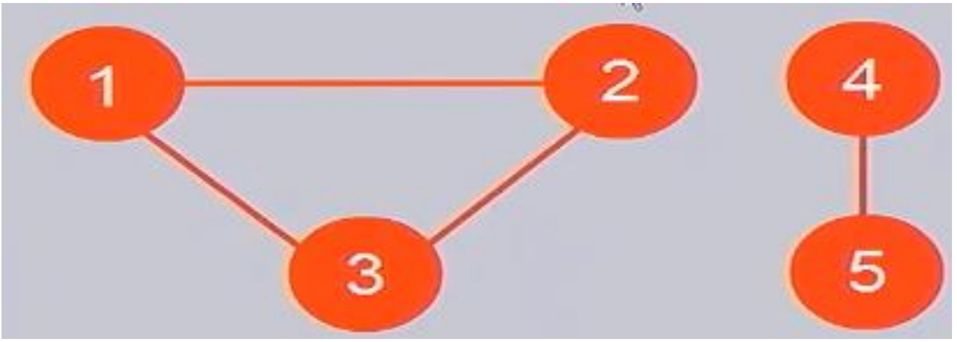
-Acyclic

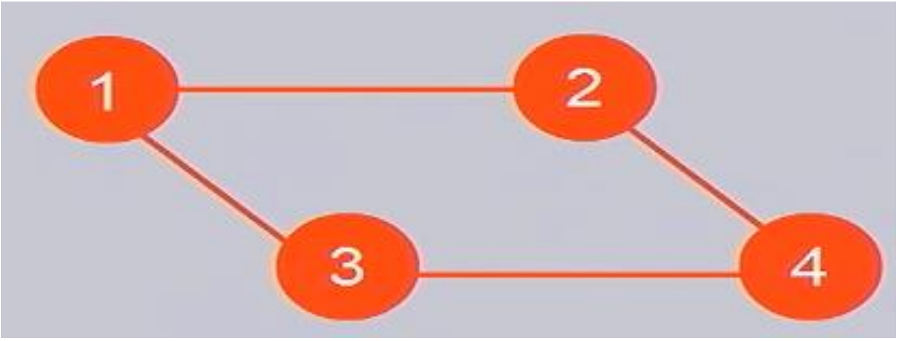
- -Undirected
- -Directed (DAG : Directed Acyclic Graph)

-Cyclic

Connectivity

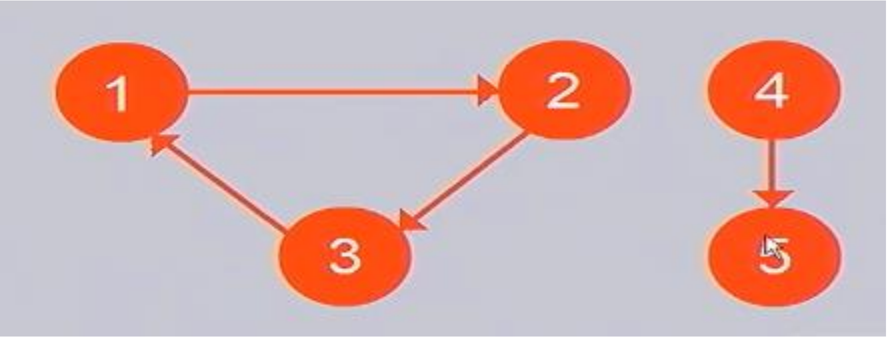
Connected if there is a path from every vertex to every vertex x.

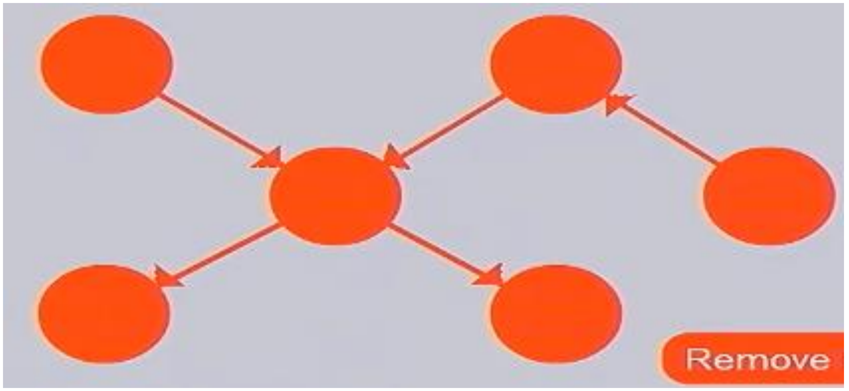




Connectivity

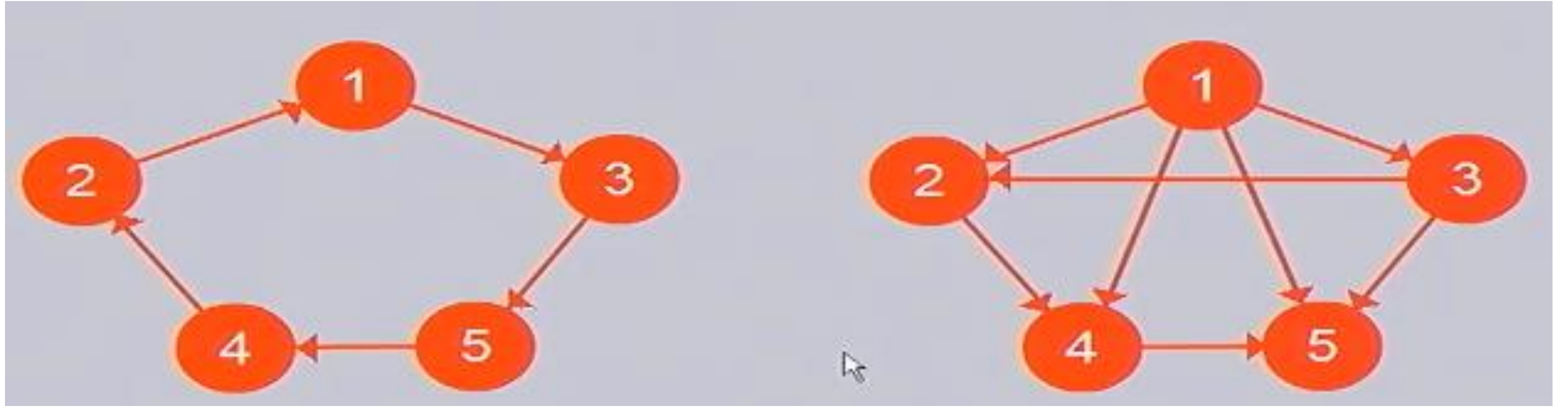
Directed



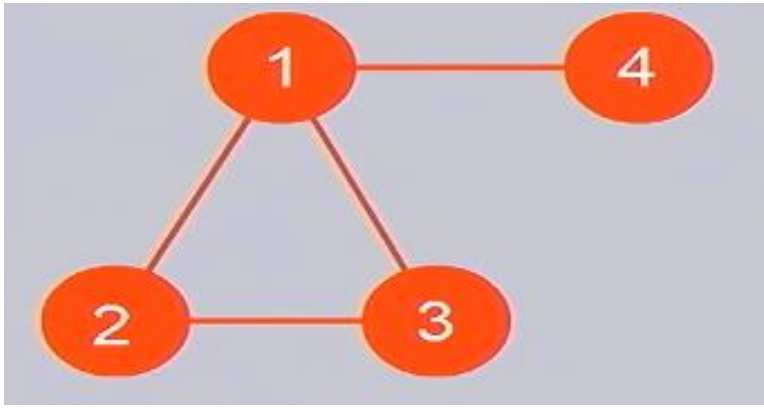


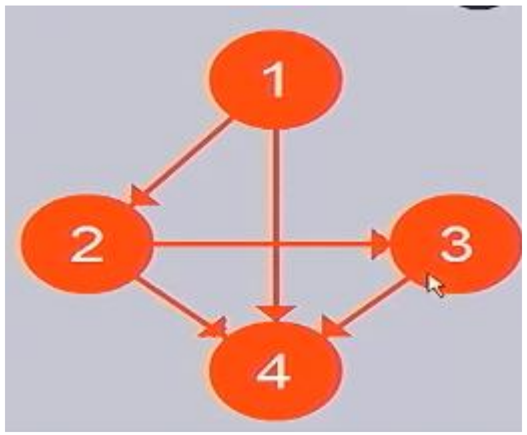
Connectivity

Strongly connected / weakly connected



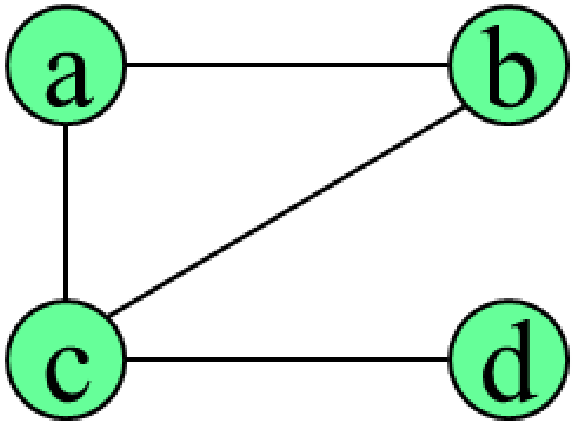
Degree





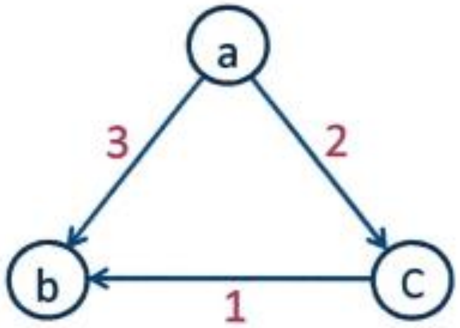
Representation of graphs

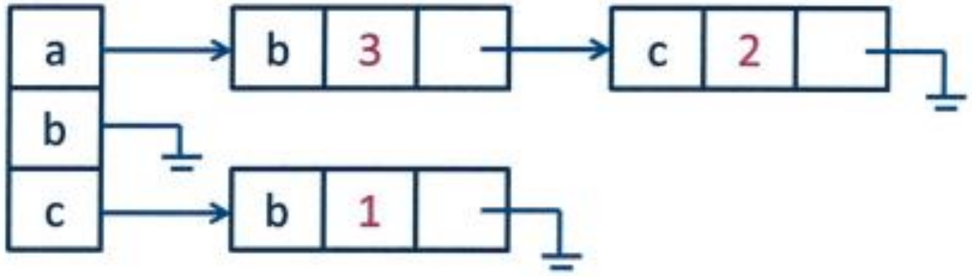
Adjacency matrix



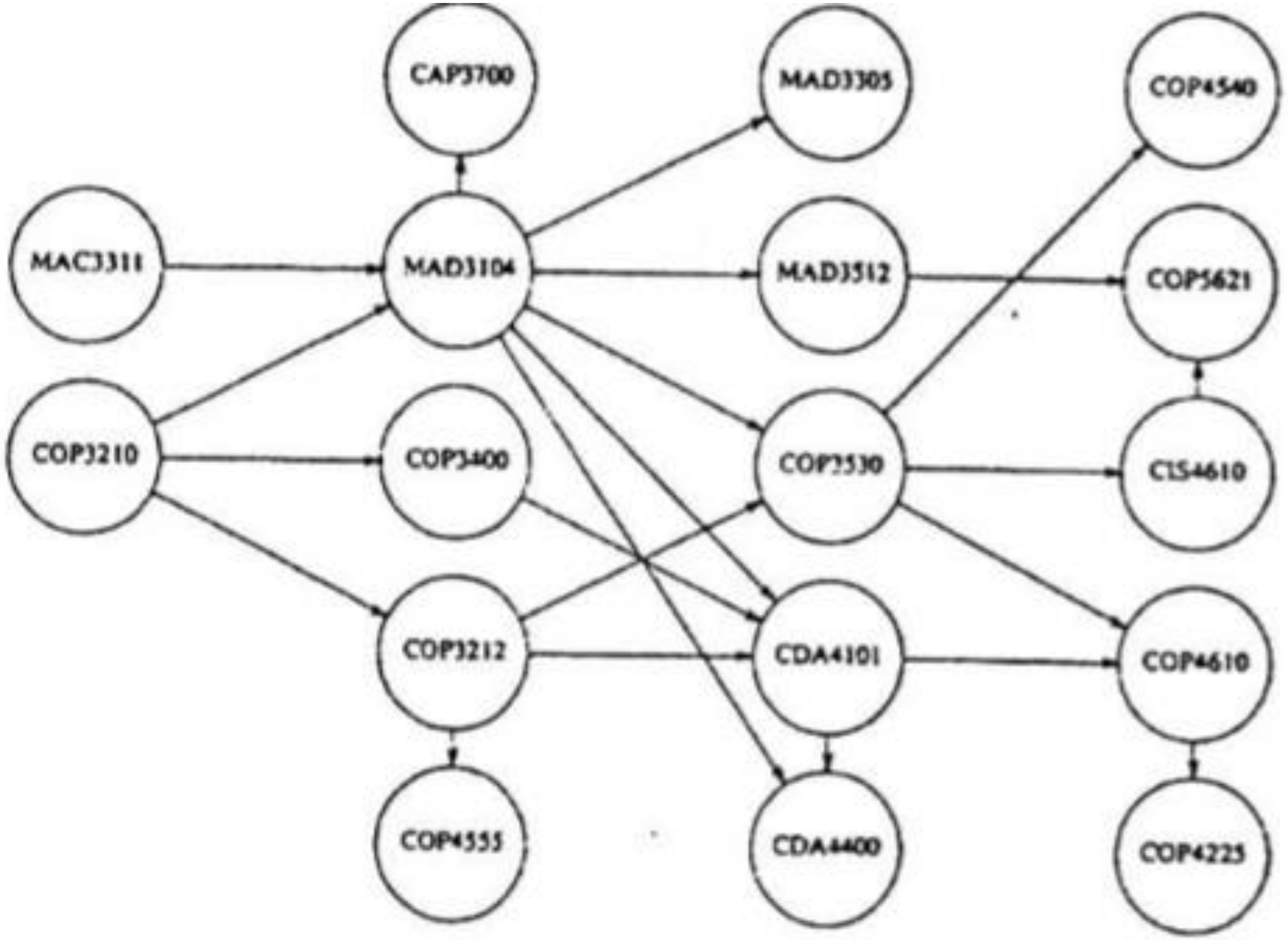
(* streets are two way. If 3000 intersections 4*n)

Adjacent list





Topological sort



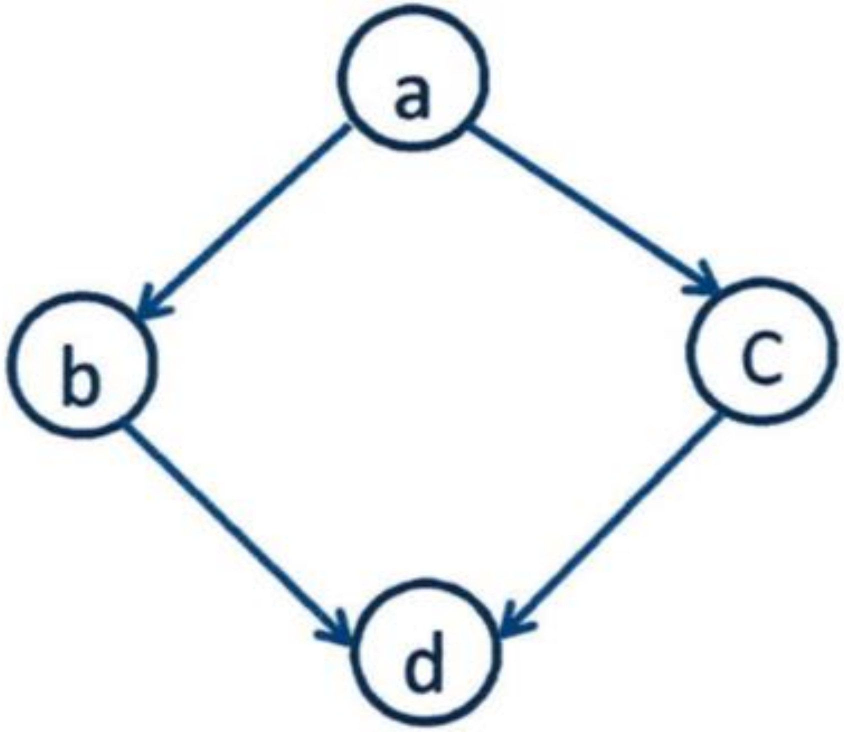
Directed

acyclic

graph

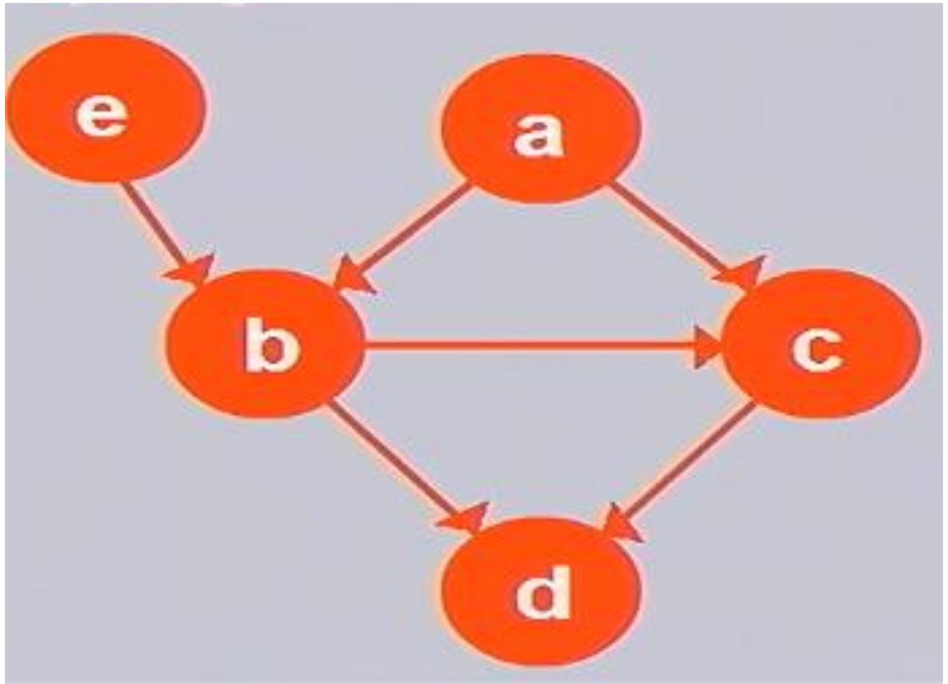
Topological sort

- Directed acyclic graph

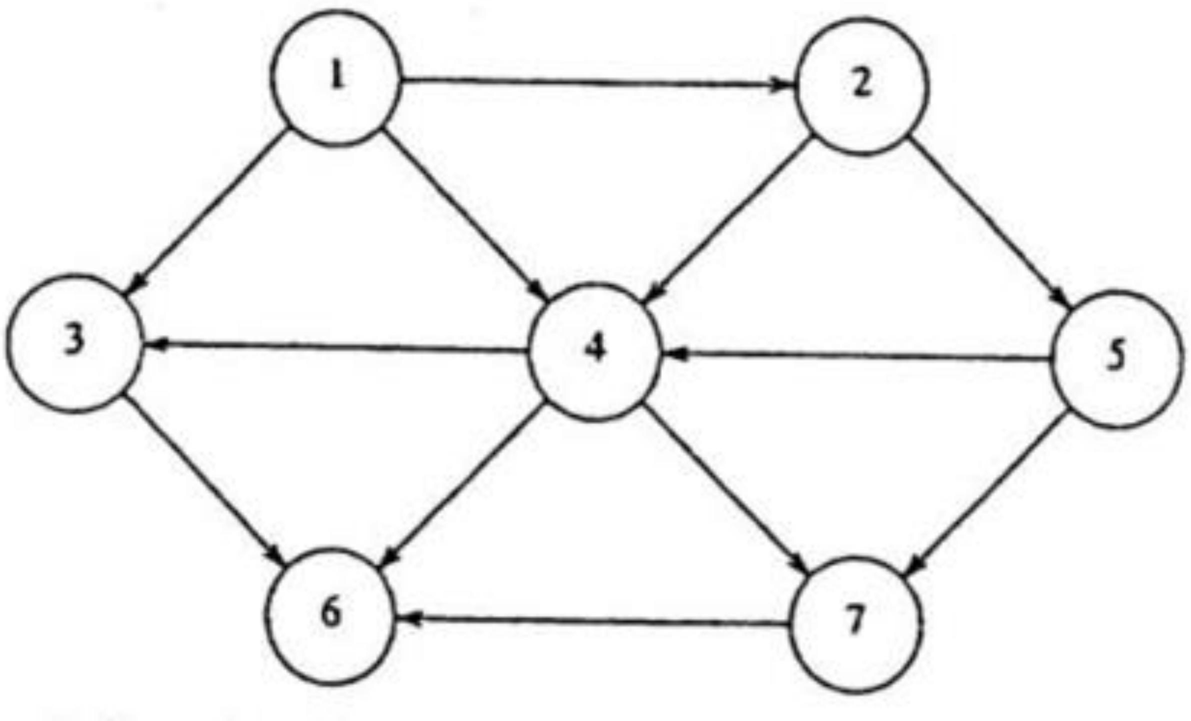


Topological sort

- Directed acyclic graph

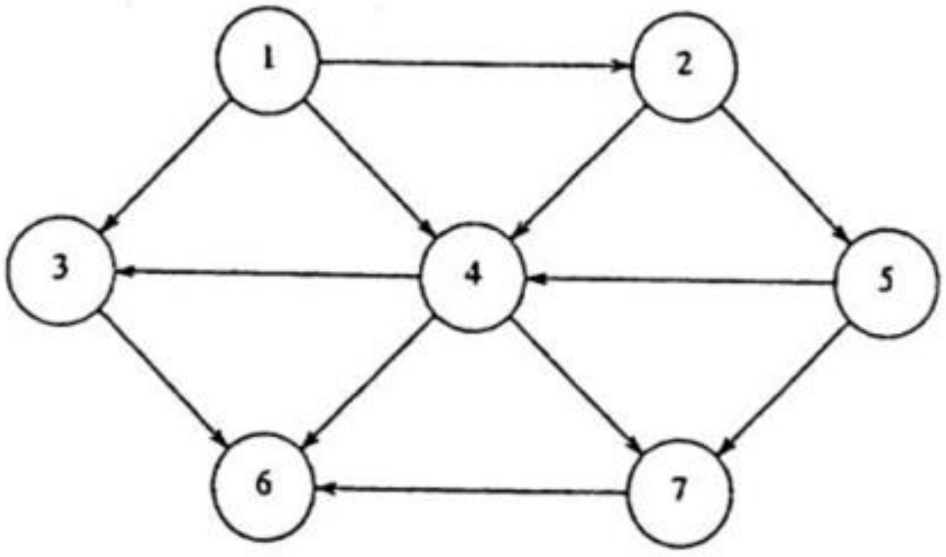


Topological sort



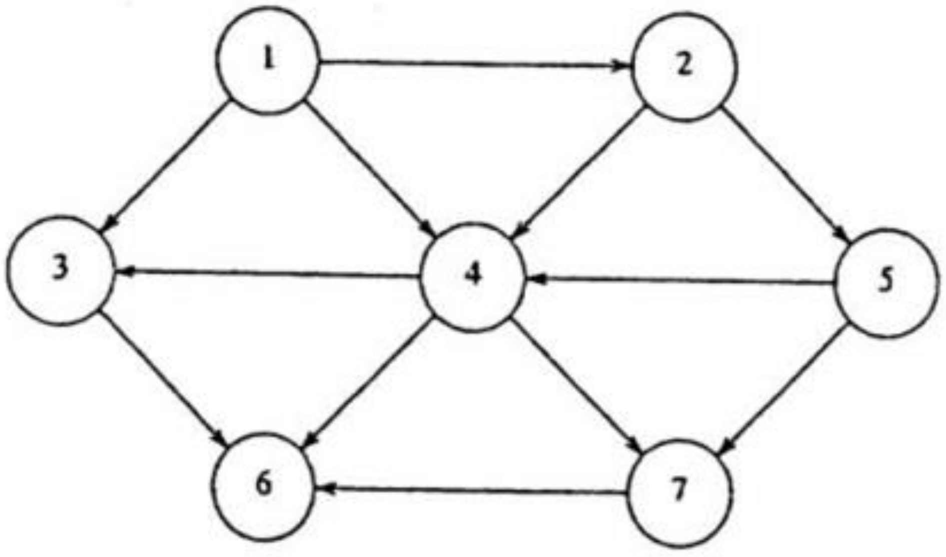
VERTEXT	INDREGREE
V1	
V2	
V3	
V4	
V5	
V6	
V7	
ENQ	
DEQ	

Topological sort



VERTEXT	INDREGREE	INDREGREE	INDREGREE
V1			
V2			
V3			
V4			
V5			
V6			
V7			
ENQ			
DEQ			

Topological sort



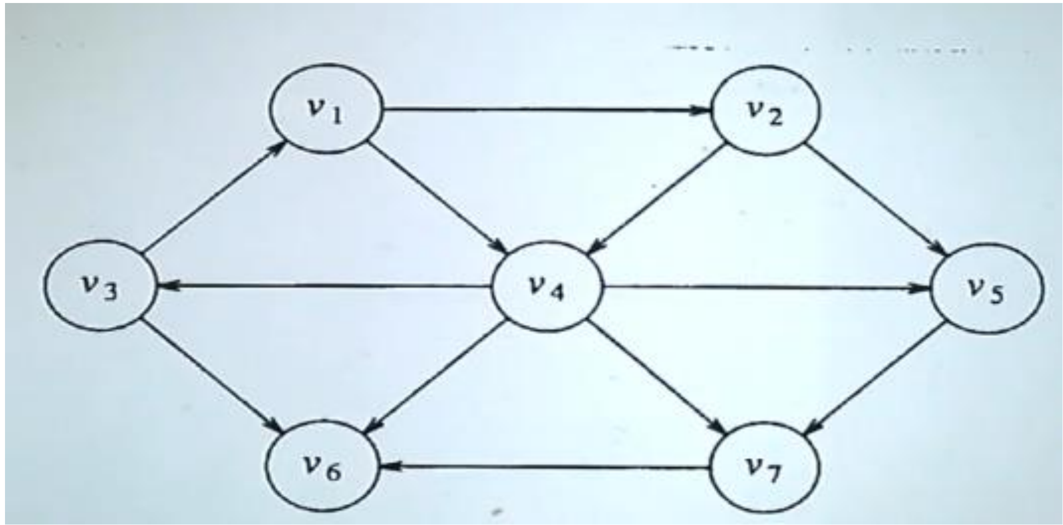
VERTEXT	INDREGREE	INDREGREE	INDREGREE
V1			
V2			
V3			
V4			
V5			
V6			
V7			
ENQ			
DEQ			

Shortest- Path algorithms

-The shortest path problem can be expressed as:

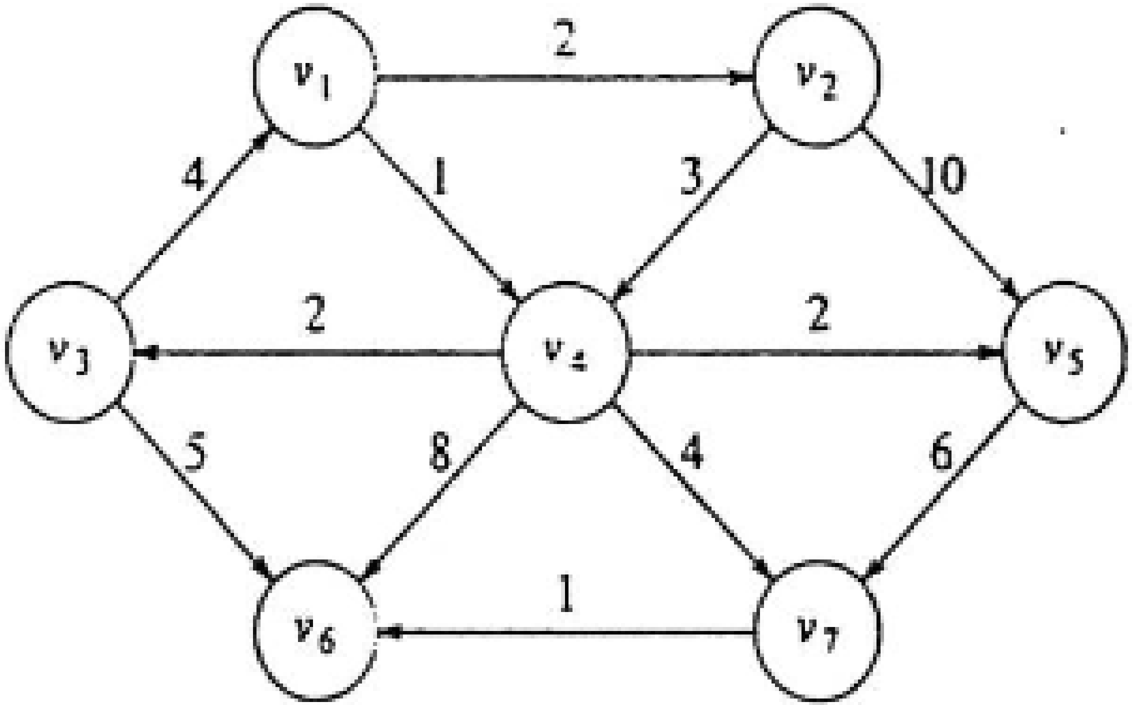
Given as input a weighted graph, G = (V,E), and a distinguished vertex,s find the shortest weighted path from s to from s to every other vertex in G

Shortest- Path : Unweighted



Vertex	Know	DV	PV
ENQ			
DEQ			

Shortest-Path:
Weighted(Dijkstra's ALG)



Vertex	Know	DV	PV
ENQ			
DEQ			