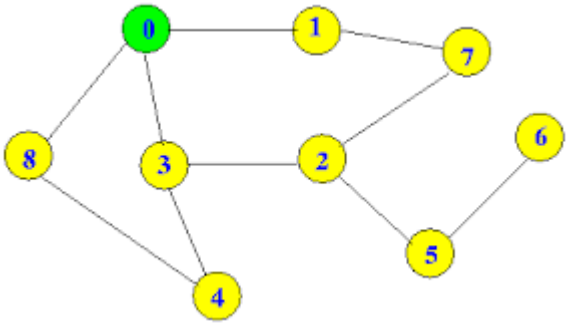
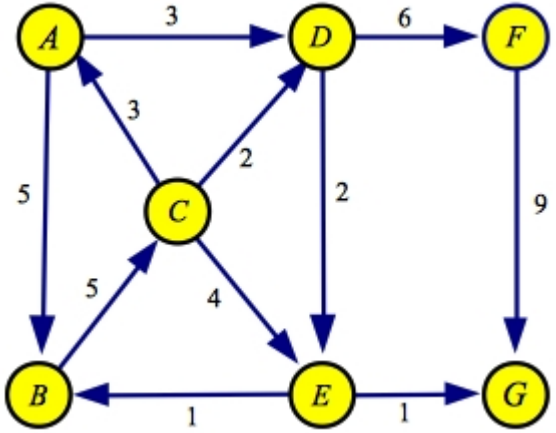
INTRODUCTION TO GRAPHS





Partha P Chakrabarti

Indian Institute of Technology Kharagpur

Graphs

A Graph G = (V, E) consists of the following:

- A set of Vertices or Nodes V
  - Nodes may have one or more labels
- A set of Edges E where each edge connects vertices of V
  - An edge usually defines a connection or relationship between vertices or nodes
  - The edges can be undirected or directed
  - Each edge can have one or more labels
  - Usually there is at most one edge between vertices, there could be multiple edges between the same nodes.
  - Normally an edge connects two vertices, but in general we could have hyper-edges

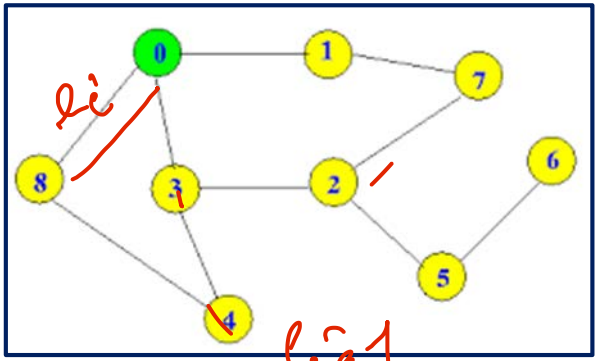


Fig 1

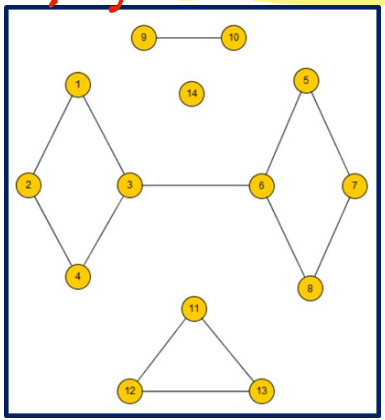


Fig 2

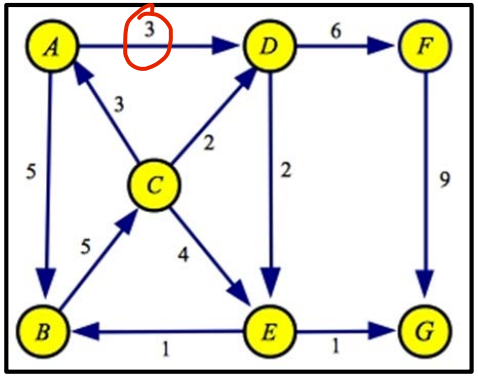


Fig 3

Graphs

A Graph G = (V, E) consists of the following:

- A set of Vertices or Nodes V
  - Nodes may have one or more labels
- A set of Edges E where each edge connects vertices of V
  - An edge usually defines a connection or relationship between vertices or nodes
  - The edges can be undirected or directed
  - Each edge can have one or more labels
  - Usually there is at most one edge between vertices, there could be multiple edges between the same nodes.
  - Normally an edge connects two vertices, but in general we could have hyper-edges

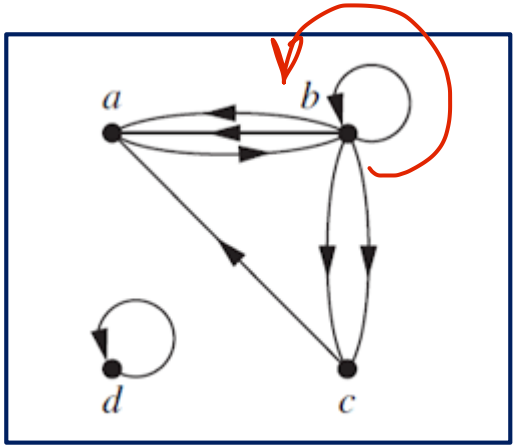
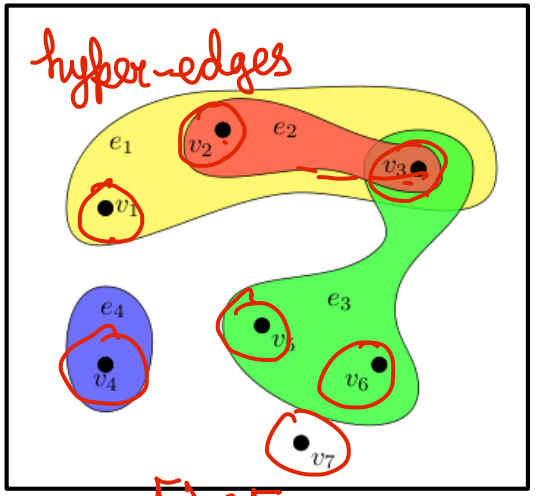


fig 4



$e = \{v_1, v_2, \ldots v_k\}$

Fig 5

Some Applications of Graphs

- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines

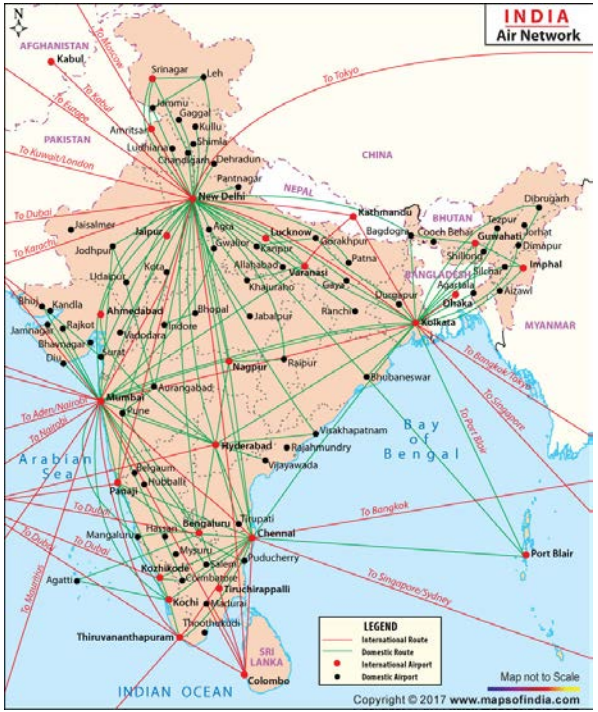
ETD ETA

airline

flight No

V = cities

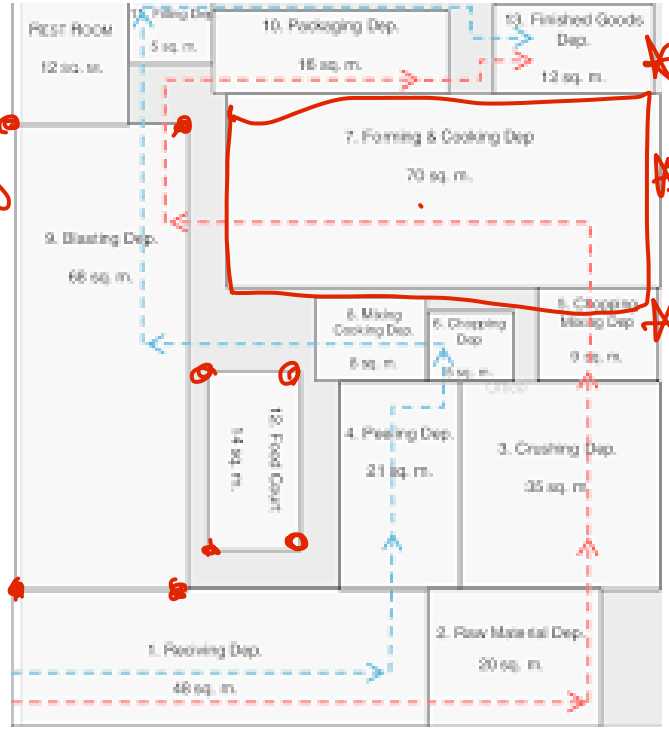
E = { set of direct flights between cities }



Appl. Eg 1

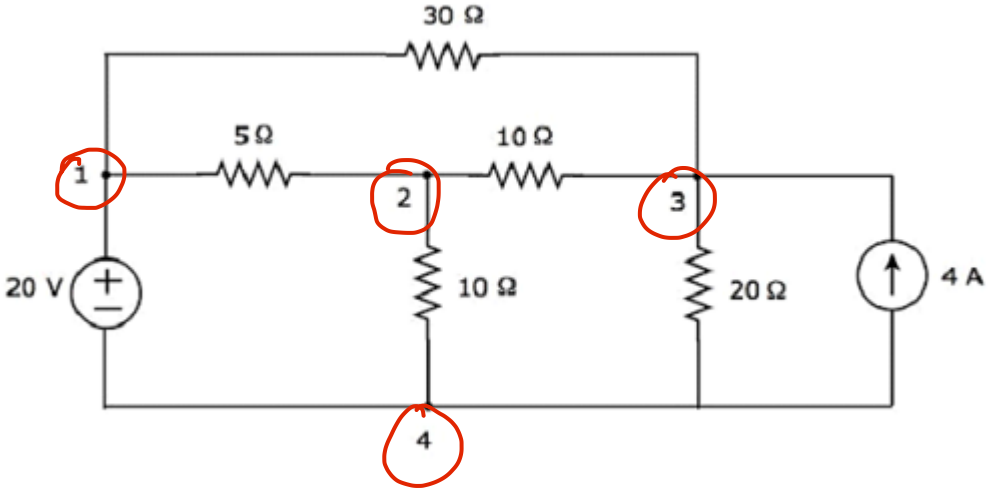
Some Applications of Graphs

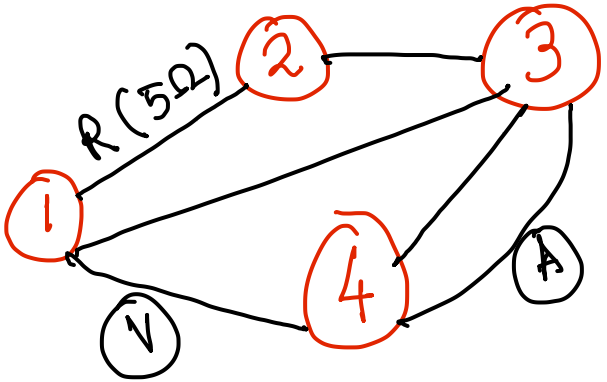
- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines



Some Applications of Graphs

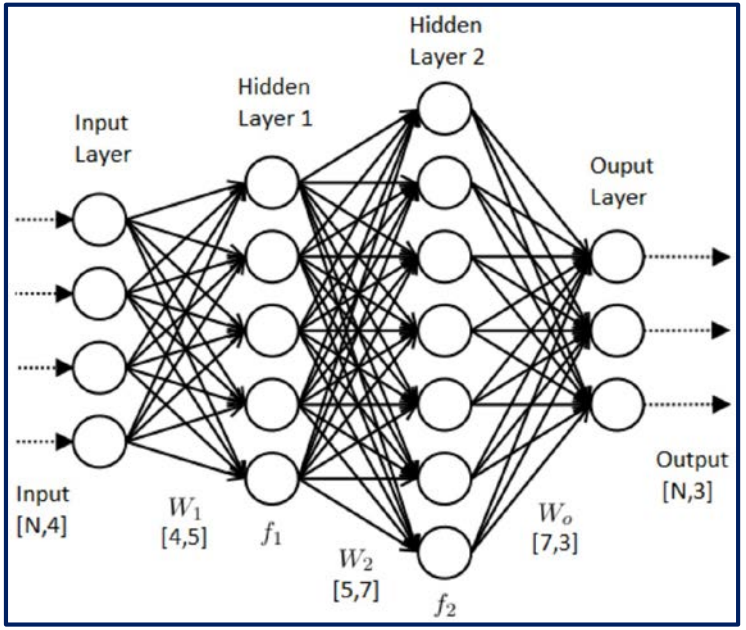
- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines





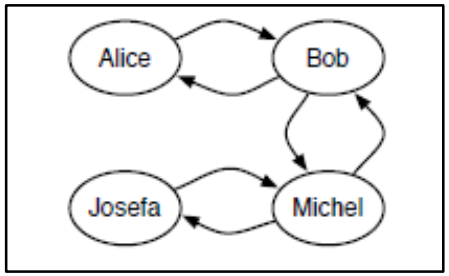
Some Applications of Graphs

- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines



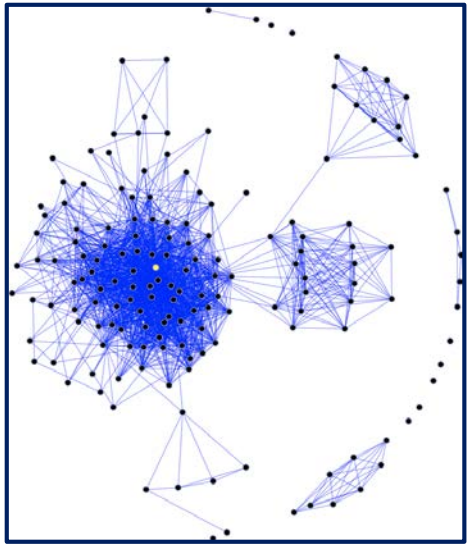
Some Applications of Graphs

- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines



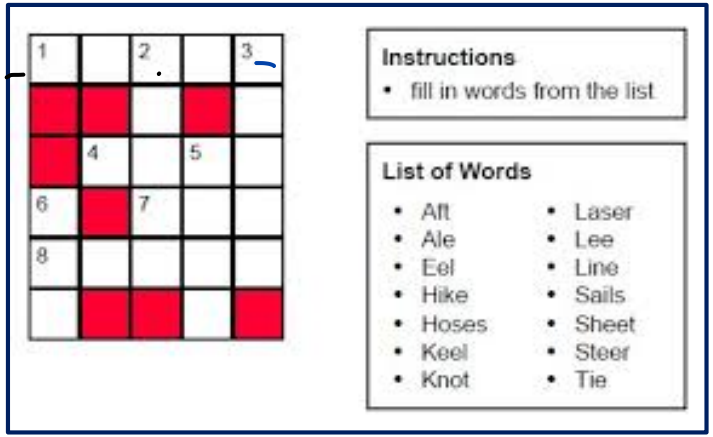
Friendship (likes)

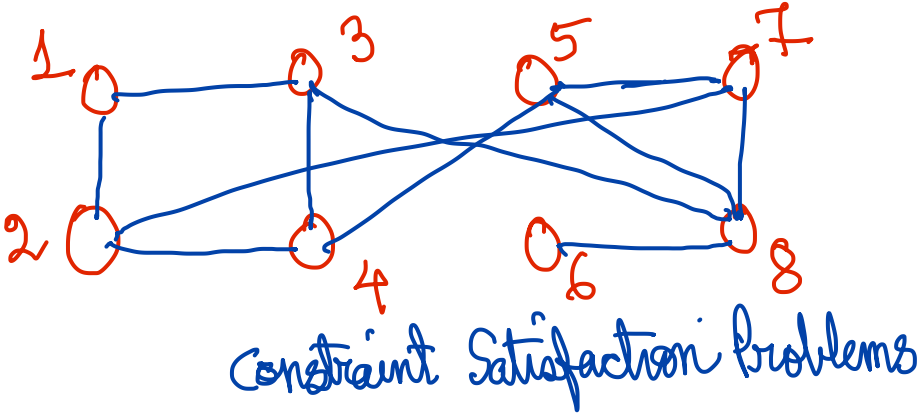
→ citation graphs



Some Applications of Graphs

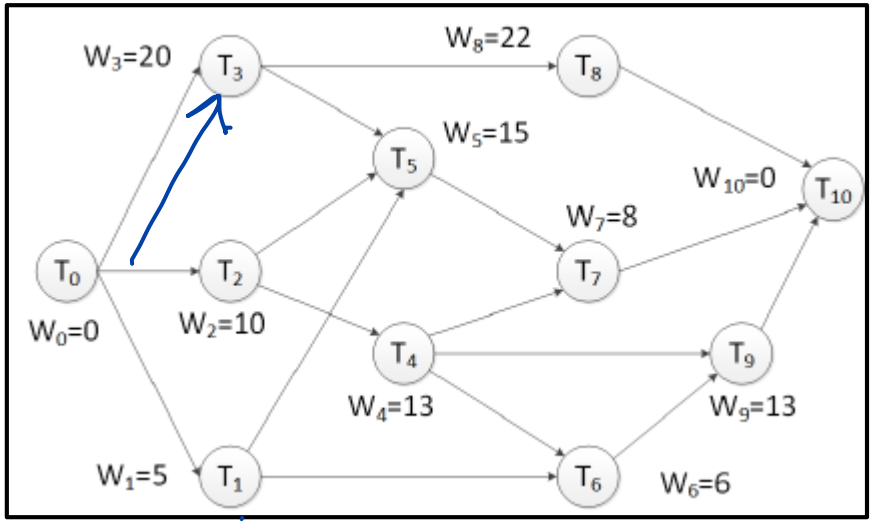
- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines





Some Applications of Graphs

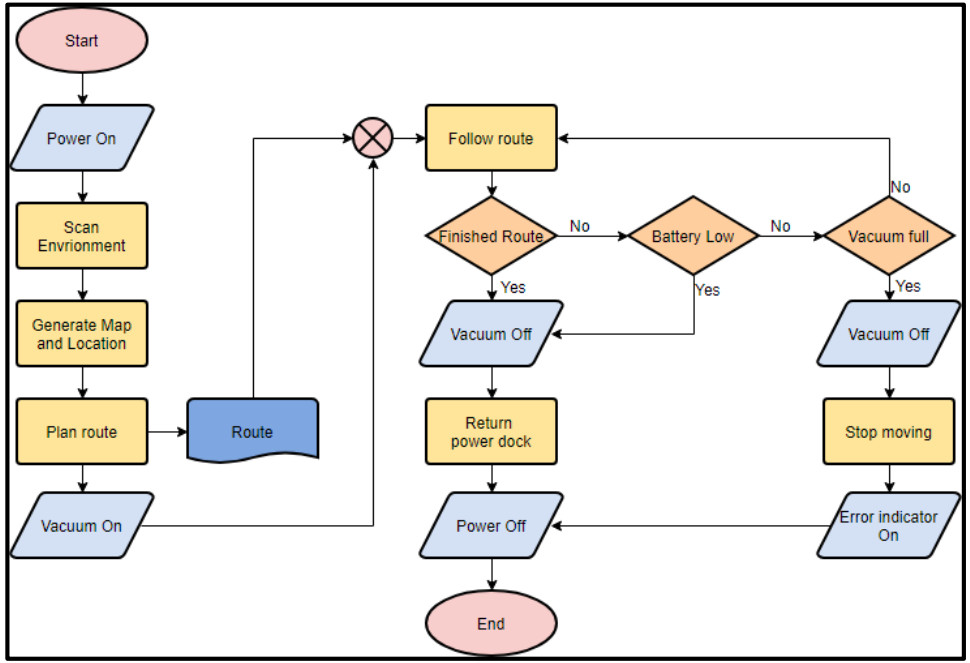
- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines



Acyclic Graphs

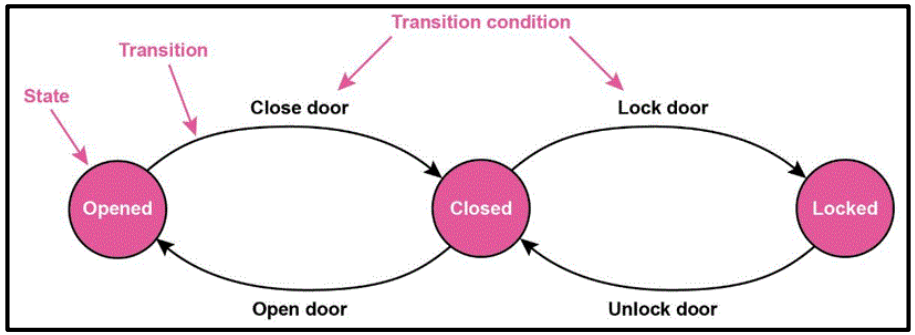
Some Applications of Graphs

- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines



Some Applications of Graphs

- Maps, Routes
- Layouts
- Circuits and Networks
- Relationships
- Constraints
- Dependencies
- Flow Charts
- State Machines



State = Node

Transition = edge

Graph Representation

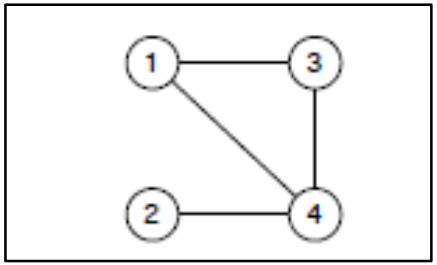


Fig 1

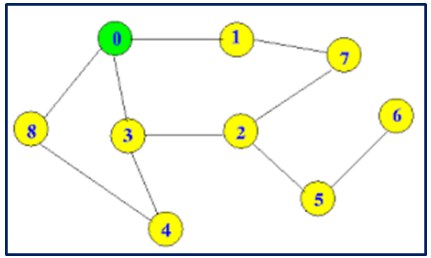
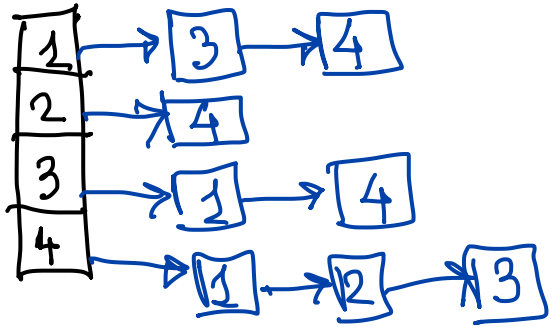


Fig 2

Adjacency Matrix

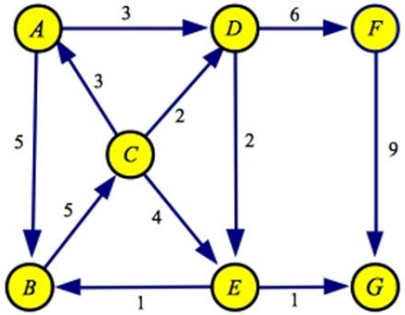
	1	2	3	4
1	0	0	1	1
2	0	0	0	1
3	1	0	0	1
4	1	1	1	0

Adjacency List

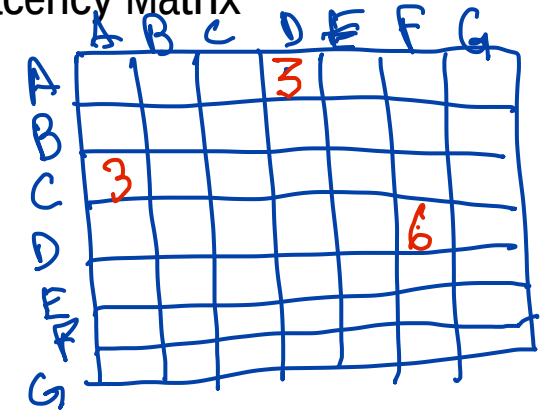


1	2	3	4	
2	1	4		
3	2	1	4	
4	3	1	2	3

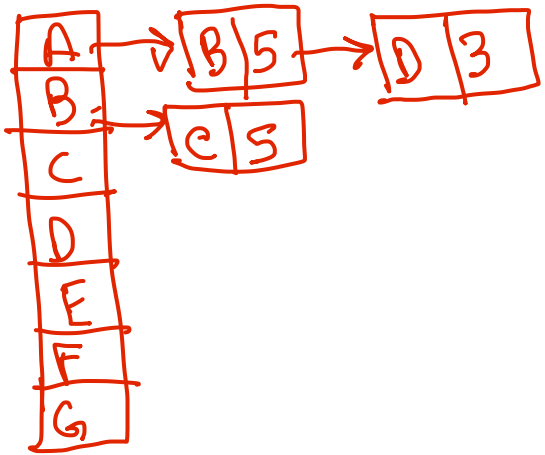
Graph Representation



Adjacency Matrix

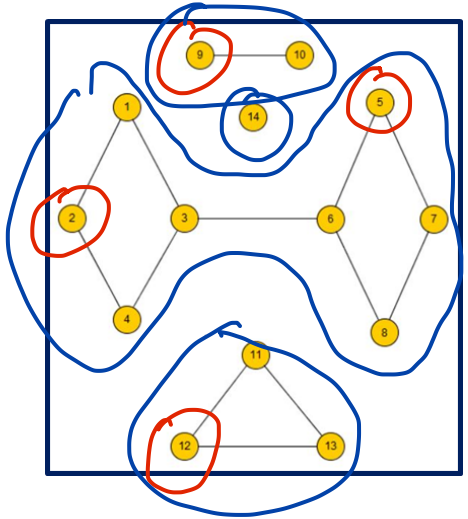


Adjacency List



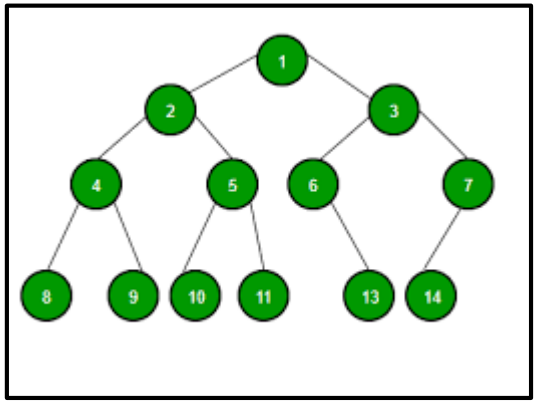
Some Algorithms on Graphs

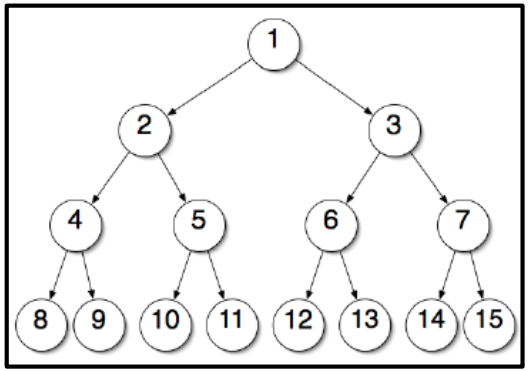
- Paths
- Reachability
- Connected Components
- Trees, Cycles, ordering
- Costs & Distances
- Spanning Trees
- Shortest Paths
- Flows

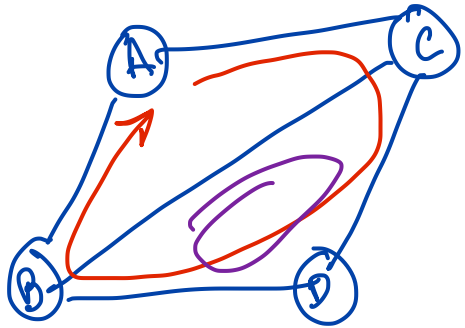


Some Algorithms on Graphs

- Paths
- Reachability
- Trees, Cycles, ordering
- Connected Components
- Costs & Distances
- Spanning Trees
- Shortest Paths
- Flows







Some Algorithms on Graphs

- Paths
- Reachability
- Cycles, ordering
- Connected Components
- Costs & Distances
- Spanning Trees
- Shortest Paths
- Flows

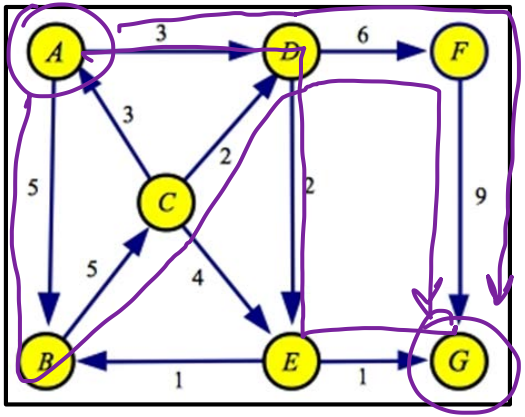


Fig 1

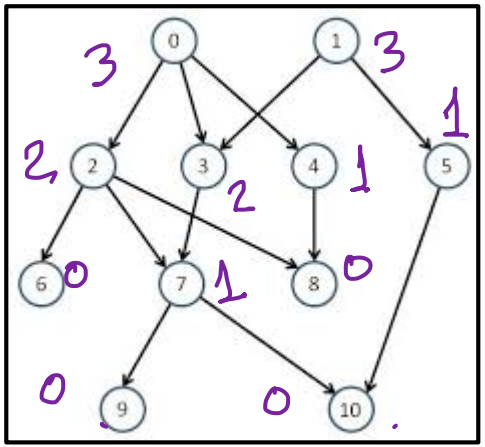
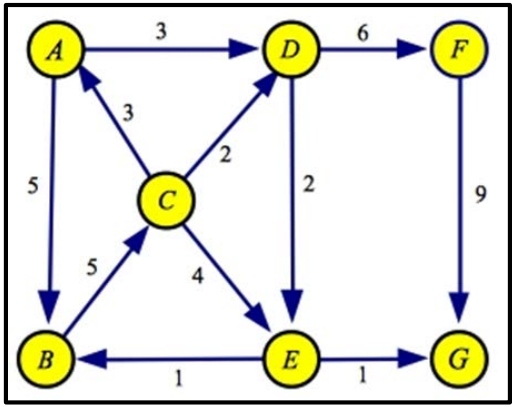
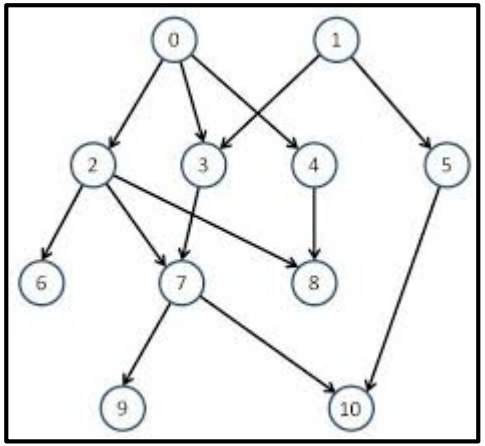


Fig 2

Some Algorithms on Graphs

- Paths
- Reachability
- Cycles, ordering
- Connected Components
- Costs & Distances
- Spanning Trees
- Shortest Paths
- Flows





Some Algorithms on Graphs

- Paths
- Reachability
- Cycles, ordering
- Connected Components
- Costs & Distances
- Spanning Trees
- Shortest Paths
- Flows

$G = (V, E)$

V = set of vertices

E = set of edges

undirected / directed

weights / labels

cycles / acyclic

TRAVERSAL OF GRAPHS

Algorithms for Graphs

Thank you