Data Structure
Chapter 5:Graphs

GATE CS Lectures

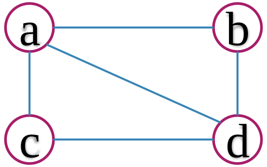
By

Monalisa Pradhan

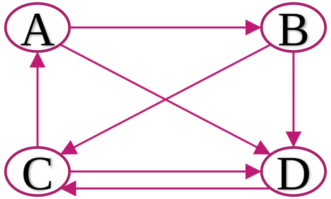
- **Section 4: Programming and Data Structures**
  Programming in C. Recursion.Arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs.

- Chapter 1:Arrays
- Chapter 2: stacks, queues
- Chapter 3: linked lists
- Chapter 4: trees
- Chapter 5: graphs(Intro,DFS,BFS)

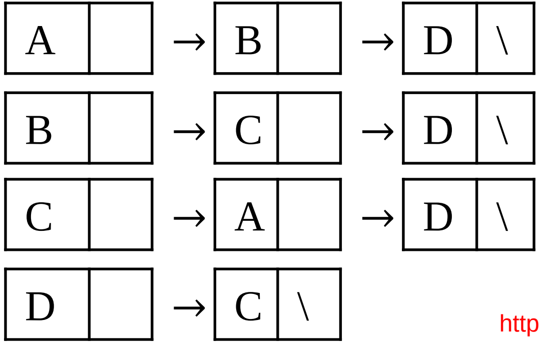
- **Graph:**
- A graph G is denoted by pair of sets G=(V,E) where V=set of all vertices in G. E=set of all edges in G.



- |V|=Number of vertices in G / Order of graph
- |E|=Number of edges in G/Size pf graph
- Non directed graph/Undirected graph: Edges without direction.
- Directed graph: Edges with direction.
- Weighted Graph: Edges with weight.
- Adjacency Matrix :It is a square matrix used to represent a graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph.
- Adjacency list :It is a linked list representation of graph.



	A	B	C	D
A	0	1	0	1
B	0	0	1	1
C	1	0	0	1
D	0	0	1	0



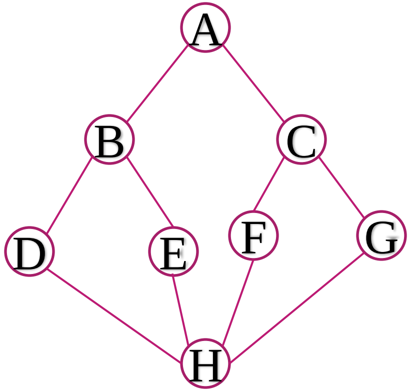
A → B → D \

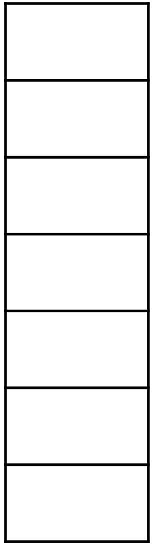
B → C → D \

C → A → D \

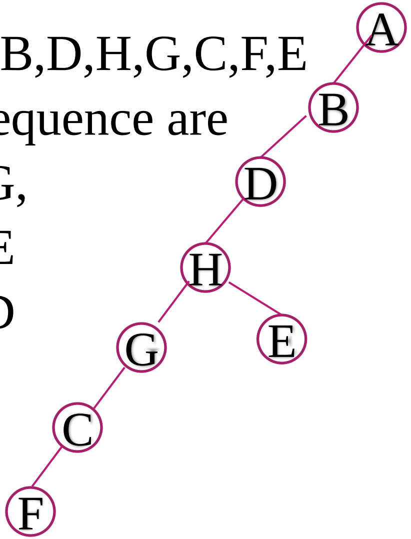
D → C \

- **Depth-First Search(**DFS)
- For the adjacency matrix representation, the traversal time is in $\boldsymbol{\Theta(|V|^2)}$, and for the adjacency list representation, it is in $\boldsymbol{\Theta(|V| + |E|)}$.
- DFS is similar to **preorder** traversal in Tree.
- It is convenient to use a **stack** to trace the operation of depth-first search.
- We push a vertex onto the stack when the vertex is reached for the first time (i.e., the visit of the vertex starts) also called **discovery time.**
- We pop a vertex off the stack when it becomes a dead end (i.e., the visit of the vertex ends) also called **finishing time.**

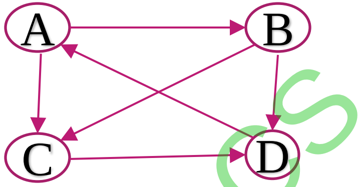


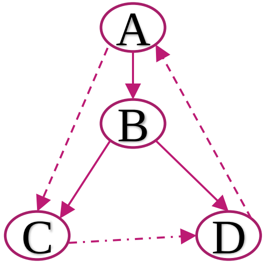


- DFS traversal 1:A,B,D,H,G,C,F,E
- Some more DFS sequence are
- 2:A,B,E,H,D,F,C,G,
- 3:H,G,C,F,A,B,D,E
- 4:A,C,F,H,G,E,B,D



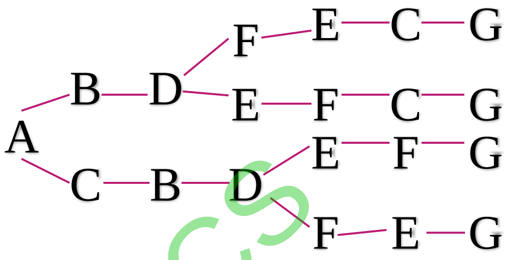
- Consider the discovery and finishing time of a 4 vertex graph find which one is connected and which one is disconnected if disconnected then how many component.
- a)(1,3)(2,2)(3,1)(4,4) b)(1,4)(2,3)(3,2)(4,1) c)(1,2)(2,1)(3,4)(4,3) d)(1,1)(2,2)(3,3)(4,4)
- a) disconnected 2 component
- b)connected
- c)disconnected 2 component
- d) disconnected 4 component

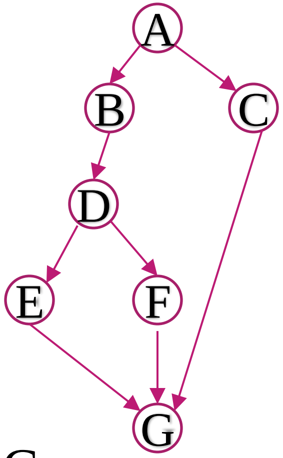




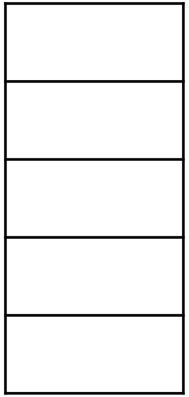
- DFS when carried out over a directed graph generate spanning tree(DFS Tree) which involve following edges
- **Tree Edge** :it is part of DFS tree. Ex: AB,BD,BC
- **Forward Edge:** It is a edge from a node to its descendent .
- which present in graph but not in DFS tree. ex:AC
- **Back edge**:It is edge from a node to its ancestor .
- which present in graph but not in DFS tree. ex:DA
- **Cross edge**:edge from a node to another node
- which is neither ancestor nor descendant ex:CD

- DFS is used for checking a graph’s **connectivity,** checking for a **cycle,** finding articulation points, Back tracking method, Topological sort.
- *Algorithm DFS(v)*
- *Visited(v)=true*
- *For each vertex(w) adjacent for ‘v’*
- *If not visited (w)*
- *Call DFS(w)*

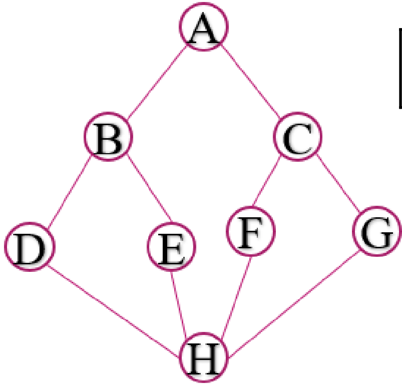


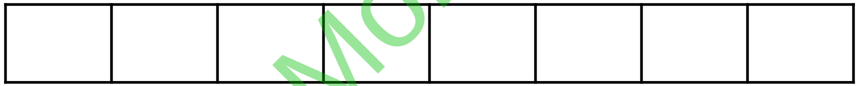


- **Topological sorting**
- Topological Sorting for a graph is not possible if the graph is not a DAG.
- It is a linear ordering of vertices such that for every directed edge uv, vertex u comes before v in the ordering.
- *Algorithm Topological sort*
- *DFS(v)*
- *Formulate a linked list of the nodes of spanning tree in decreasing order of finishing time or pop time.*



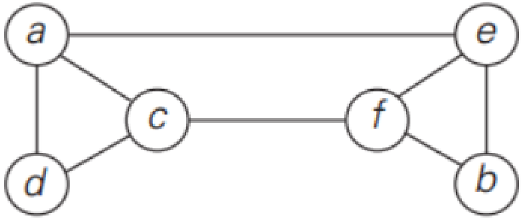
- **Breadth-First Search**
- Breadth-first search has the same efficiency as depth-first search: it is in $\Theta(|V|^2)$ for the adjacency matrix representation and in $\Theta(|V| + |E|)$ for the adjacency list representation.
- BFS is similar to level order traversal of Tree .
- It is convenient to use a queue to trace the operation of breadth-first search.
- The queue is initialized with the traversal's starting vertex, which is marked as visited.
- On each iteration, the algorithm identifies all unvisited vertices that are adjacent to the front vertex , marks them as visited, and adds them to the queue; after that, the front vertex is removed from the queue.

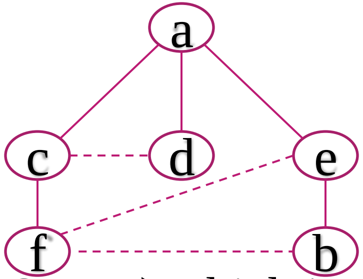




- BFS traversal 1:A,B,C,D,E,F,G,H
- Some more BFS sequence are
- 2:A,C,B,F,G,D,E,H
- 3:A,B,C,E,D,G,F,H

- BFS can be used to check connectivity and acyclicity of graph.
- It is used in branch and bound method.
- Also used for finding shortest path from root all other node.





- BFS when carried out over a graph generate spanning tree(BFS Tree) which involve following edges
- **Tree Edge** :it is part of BFS tree. Ex: ac,ad,ae,cf,be
- **Cross edge**:edge from a node to another node which are not tree edge .Ex:cd,ef,bf

	DFS	**BFS**
Data structure	Stack	Queue
Number of vertex ordering	2	1
Tree traversal	PreOrder	Level Order
Function call	recursive	Iterative
Efficiency	$\Theta(\lvert V\rvert^2)$, $\Theta(\lvert V\rvert + \lvert E\rvert)$	$\Theta(\lvert V\rvert^2)$, $\Theta(\lvert V\rvert + \lvert E\rvert)$