Graph Algorithms: Depth-First Search (DFS)

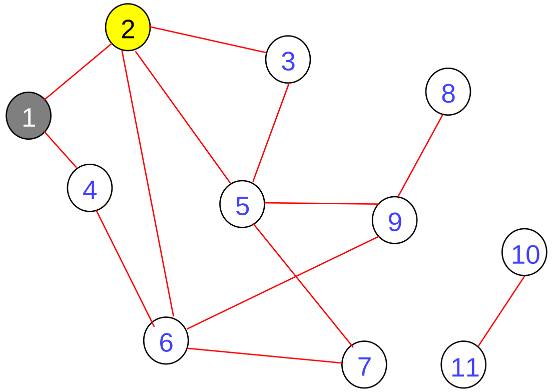
Dr. Shrutilipi Bhattacharjee, Assistant Professor, NIT Karnataka, India

Depth-first Search (DFS)

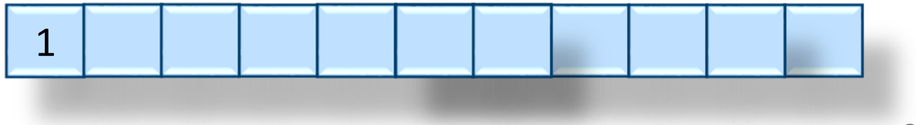
- Visit start vertex and put into a LIFO stack
- Repeatedly remove a vertex from the stack, visit its unvisited adjacent vertices, put newly visited vertices into the stack

- Start search at vertex ***1***
- Label vertex ***1*** and do a depth first search from either ***2*** or ***4***
- Suppose that vertex ***2*** is selected

```
depthFirstSearch(v)
{
  Label vertex v as discovered
  for (each undiscovered vertex u adjacent from v) do
    if vertex u is not labeled as discovered then
       depthFirstSearch(u);
}
```

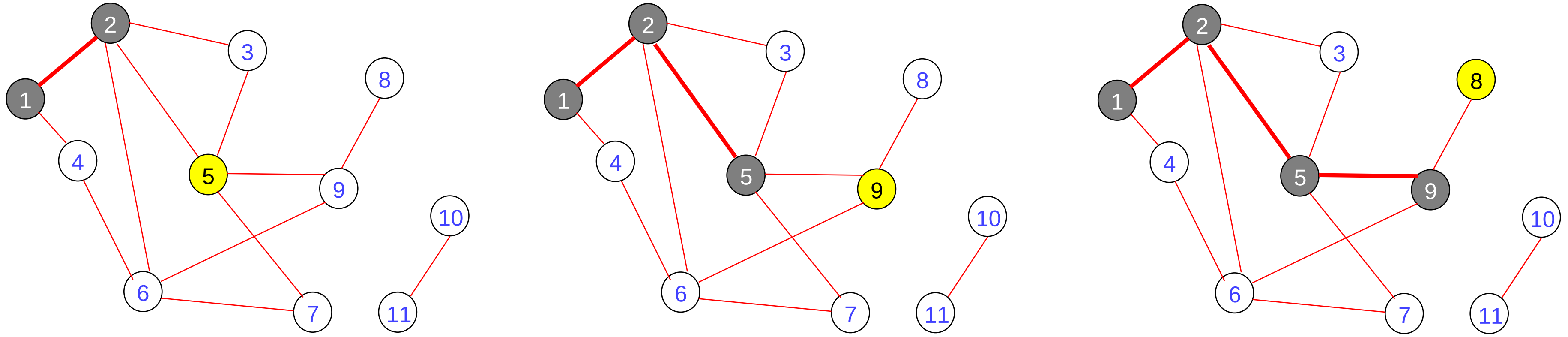


Visit/mark/label start vertex and put in a LIFO stack ***S***



Depth-first Search (DFS)

- Label vertex ***2*** and do a depth first search from either ***3***, ***5***, or ***6***
- Suppose that vertex ***5*** is selected
- Label vertex ***5*** and do a depth first search from either ***3***, ***7***, or ***9***
- Suppose that vertex ***9*** is selected
- Label vertex ***9*** and do a depth first search from either ***6*** or ***8***
- Suppose that vertex ***8*** is selected

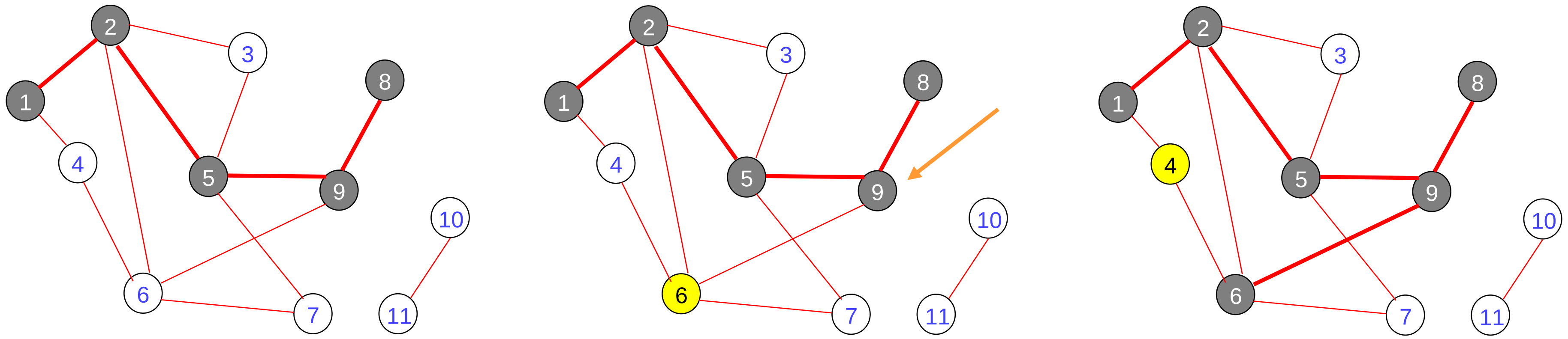


Visit/mark/label start vertex and put in a LIFO Stack ***S***

1	2	5	9	8						

Depth-first Search (DFS)

- Label vertex ***8*** and return to vertex ***9***
- From vertex ***9*** and do a depthFirstSearch(***6***)
- Label vertex ***6*** and do a depth first search from either ***4*** or ***7***
- Suppose that vertex ***4*** is selected

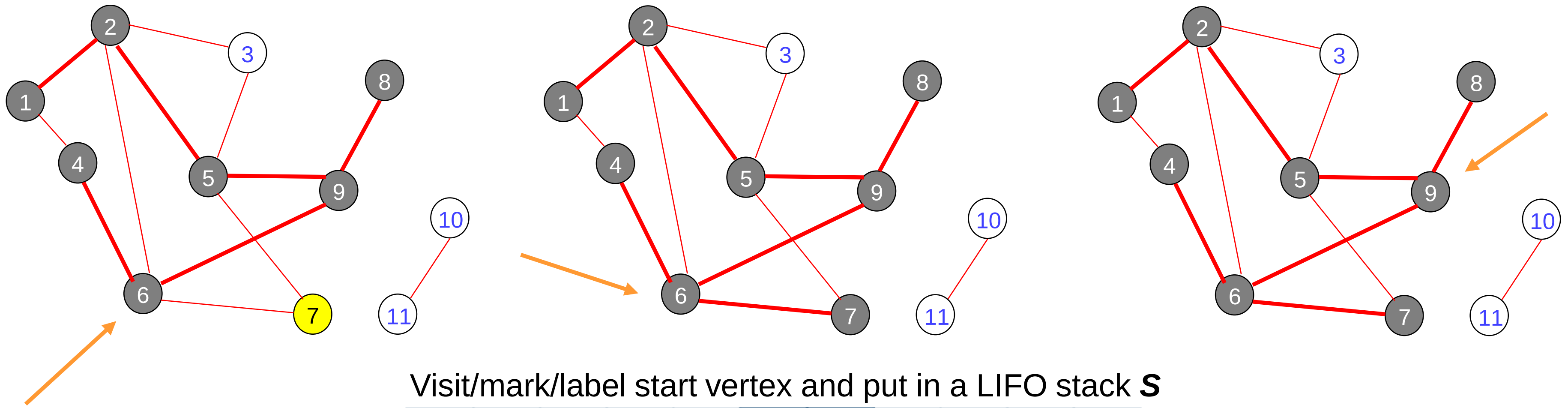


Visit/mark/label start vertex and put in a LIFO stack ***S***

1	2	5	9	8	6	4				

Depth-first Search (DFS)

- Label vertex ***4*** and return to vertex ***6***
- From vertex ***6*** and do a depthFirstSearch(***7***)
- Label vertex ***7*** and return to vertex ***6***
- Return to ***9***

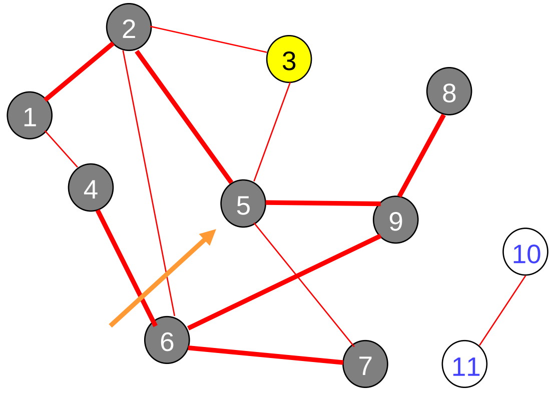


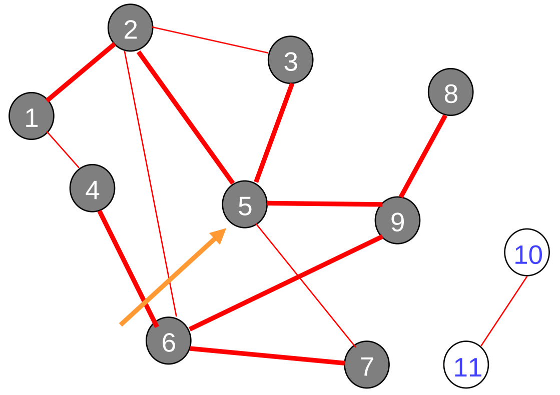
Visit/mark/label start vertex and put in a LIFO stack ***S***

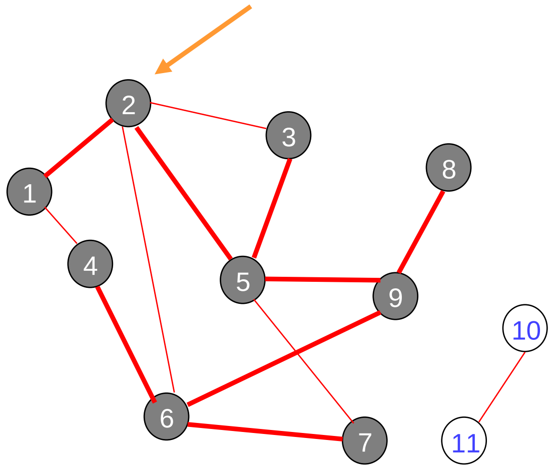
1	2	5	9	8	6	4	7			

Depth-first Search (DFS)

- Return to ***5*** and do a depthFirstSearch(***3***)
- Label vertex ***3*** and return to vertex ***5***
- Return to ***2***





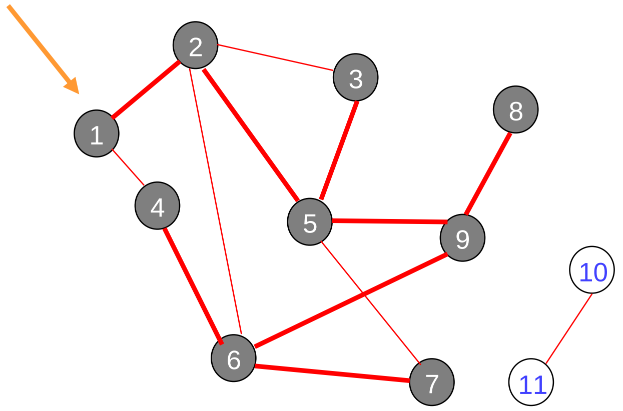


Visit/mark/label start vertex and put in a LIFO stack ***S***

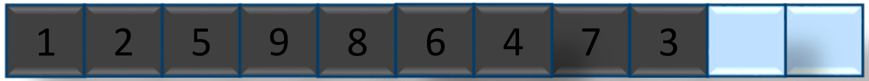
1	2	5	9	8	6	4	7	3		

Depth-first Search (DFS)

- Return to ***1***
- Return to the invoking method



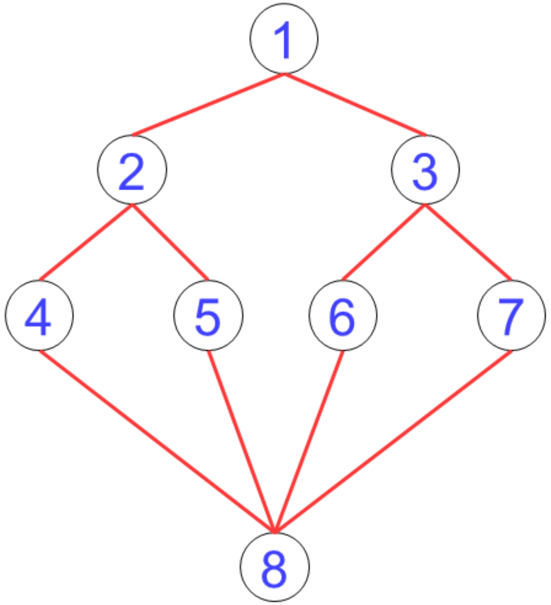
Visit/mark/label start vertex and put in a LIFO stack ***S***



1	2	5	9	8	6	4	7	3		

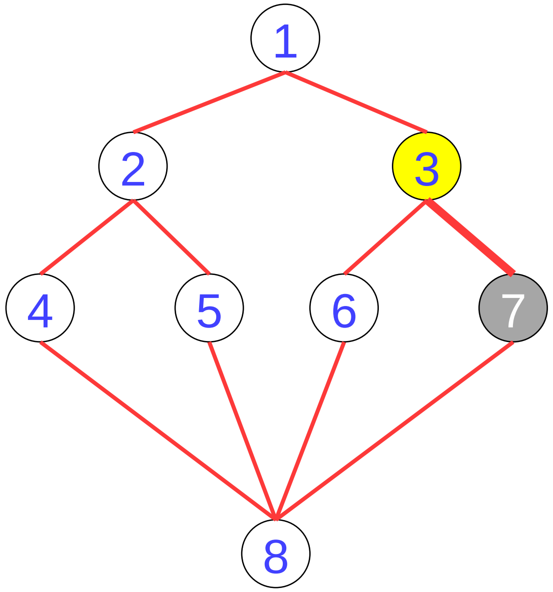
Depth-first Search (DFS)

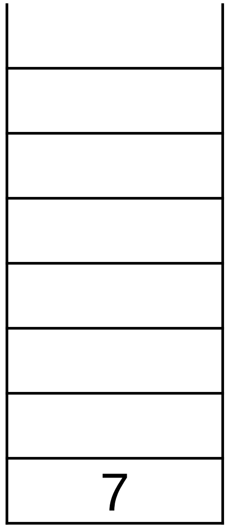
- Traverse the graph using depth-first-search algorithm
- Consider the starting vertex to be vertex **7**
- Clearly write all the intermediate changes in the data structure used



Depth-first Search (DFS)

- For the DFS traversal, a stack of size **8** is initialized below:
- Start search at vertex **7**
- Label vertex **7** and do a depth first search from either **3** or **8**
- Suppose vertex **3** is selected

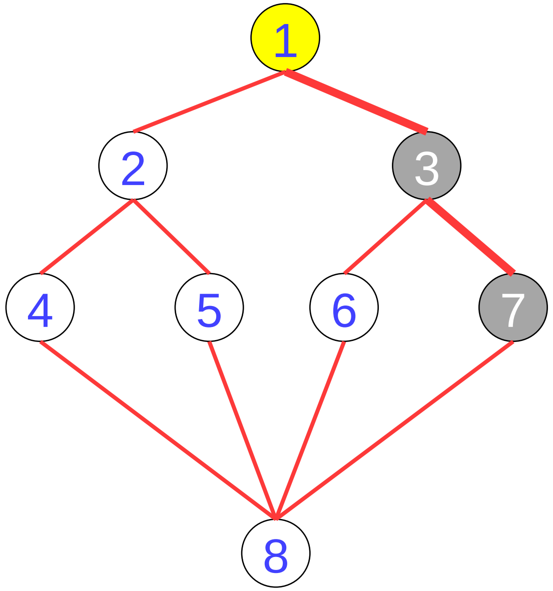


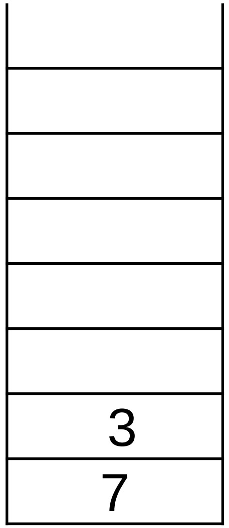


The DFS stack

Depth-first Search (DFS)

- Label vertex **3** and do a depth first search from either **1** or **6**
- Suppose vertex **1** is selected

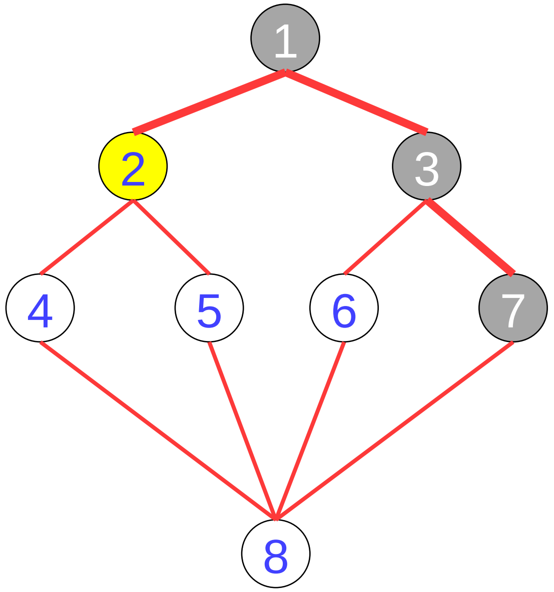


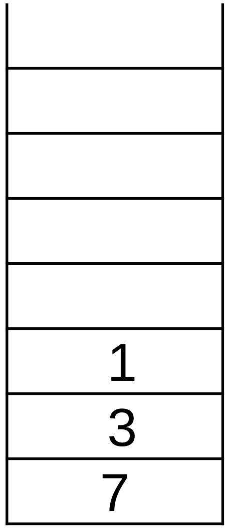


The DFS stack

Depth-first Search (DFS)

- Label vertex **1** and do a depth first search from **2**

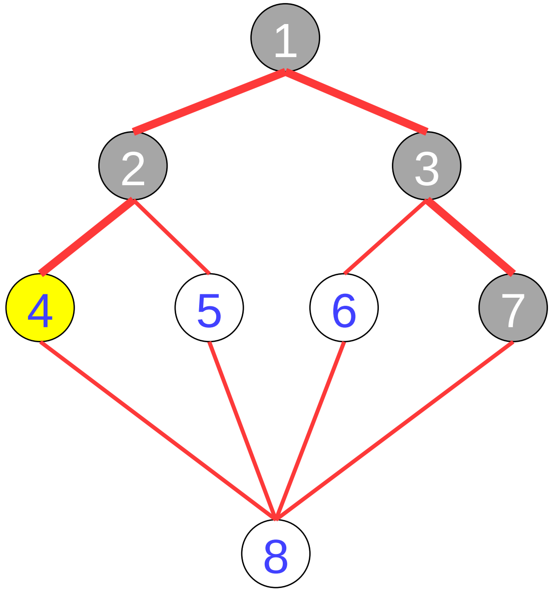


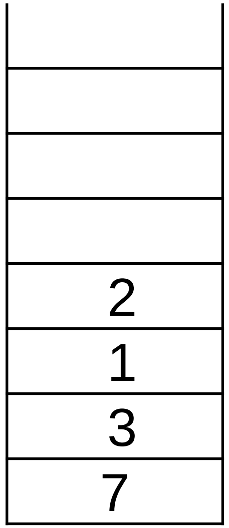


The DFS stack

Depth-first Search (DFS)

- Label vertex **2** and do a depth first search from either **4** or **5**
- Suppose vertex **4** is selected

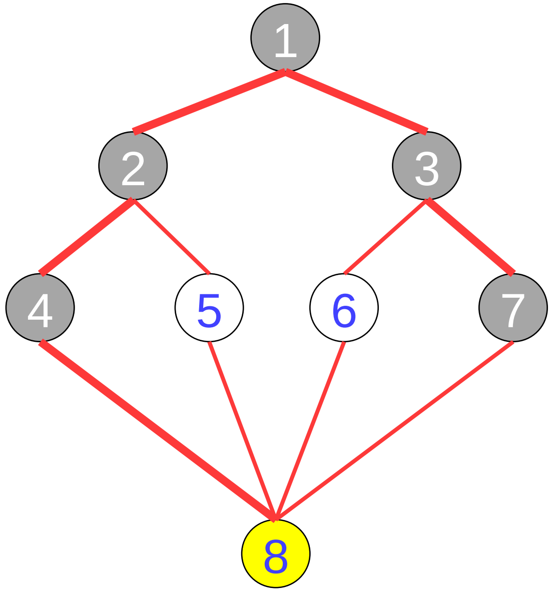


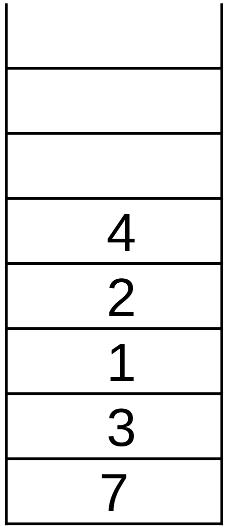


The DFS stack

Depth-first Search (DFS)

- Label vertex **4** and do a depth first search from **8**

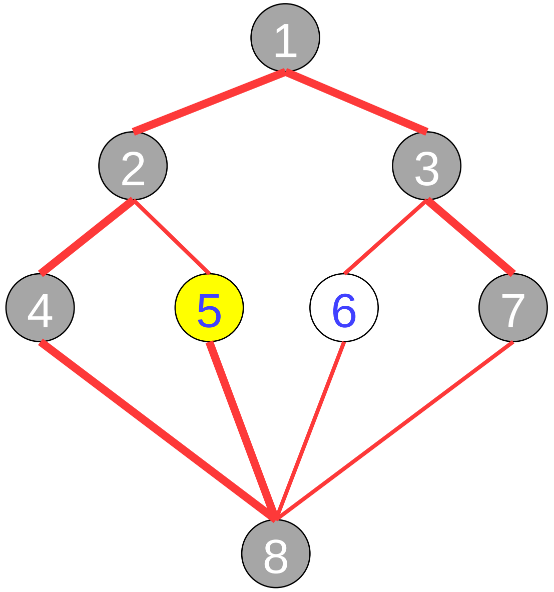


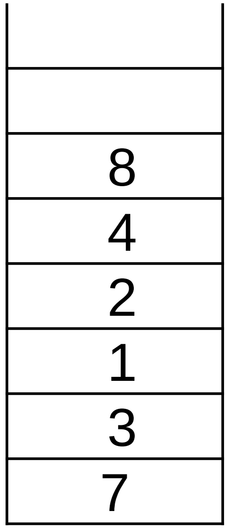


The DFS stack

Depth-first Search (DFS)

- Label vertex **8** and do a depth first search from either **5** or **6**
- Suppose vertex **5** is selected

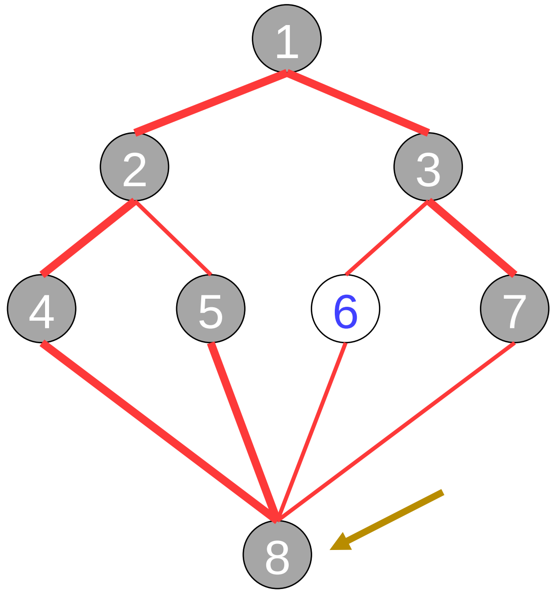


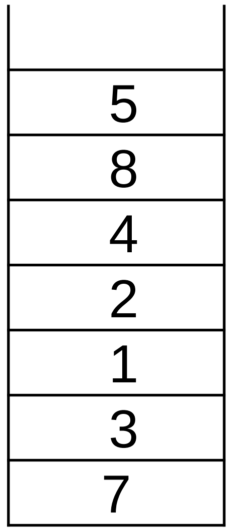


The DFS stack

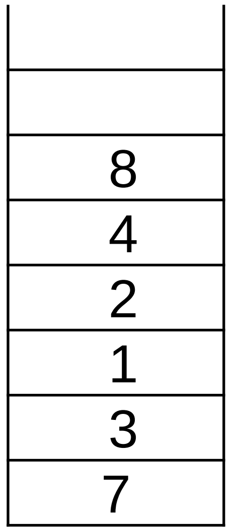
Depth-first Search (DFS)

- Label vertex **5** and return to vertex **8**





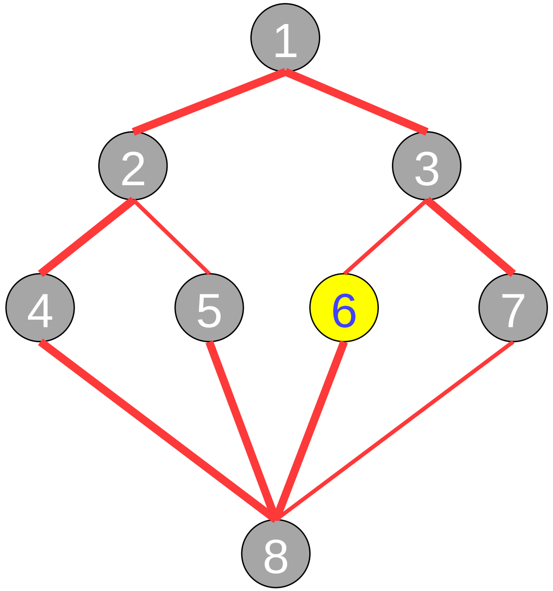
The DFS stack

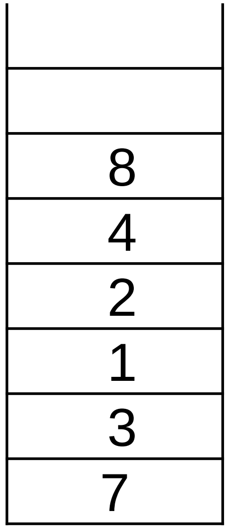


The DFS stack

Depth-first Search (DFS)

- Start search at vertex **8** and do a depth first search from **6**

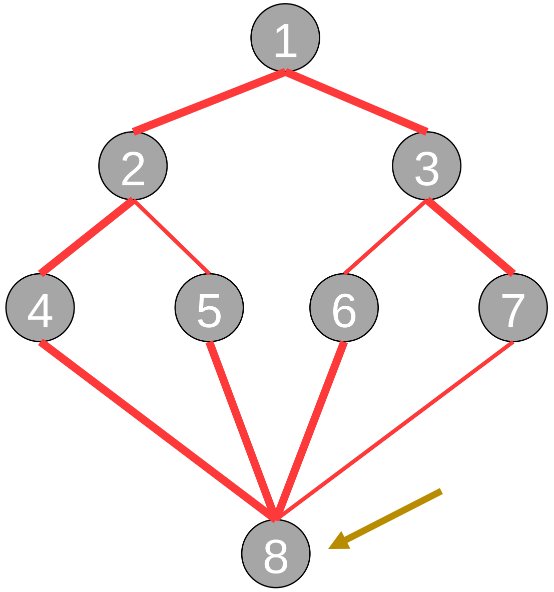


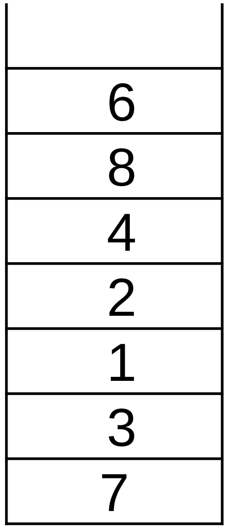


The DFS stack

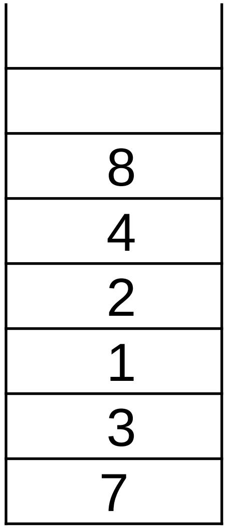
Depth-first Search (DFS)

- Label vertex 6 and return to vertex 8





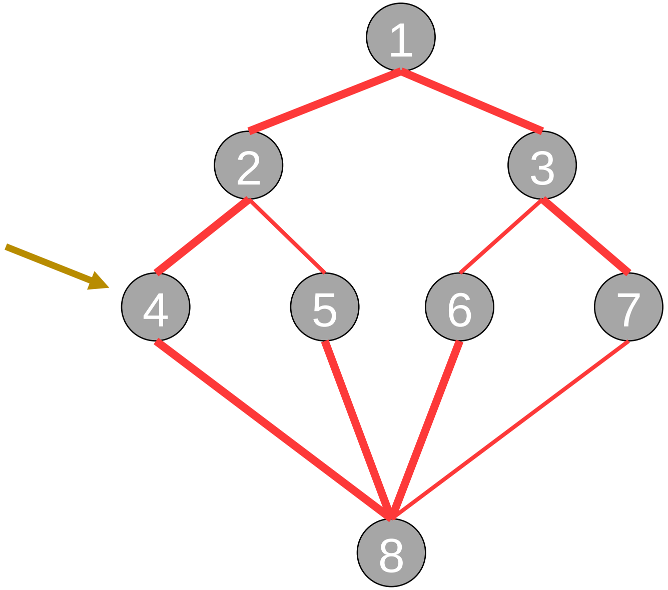
The DFS stack

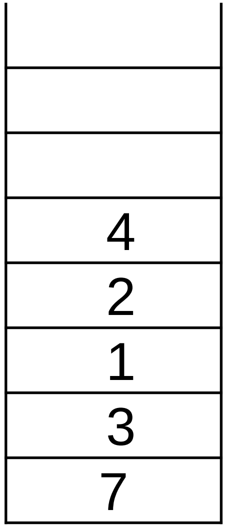


The DFS stack

Depth-first Search (DFS)

- Return to vertex **4**

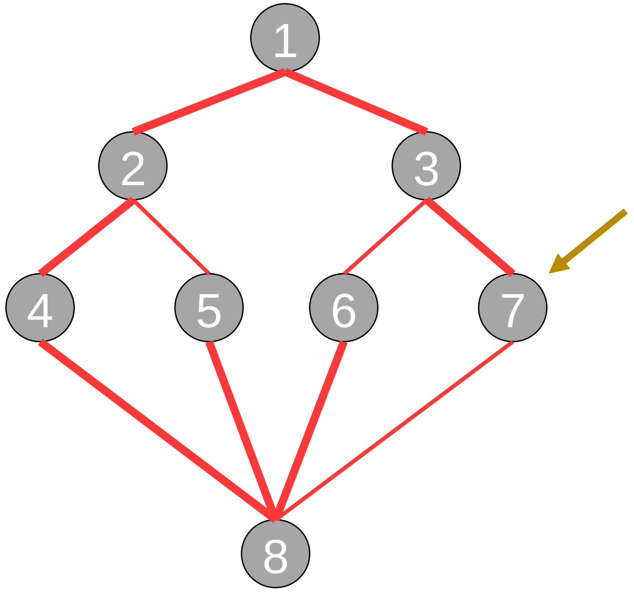


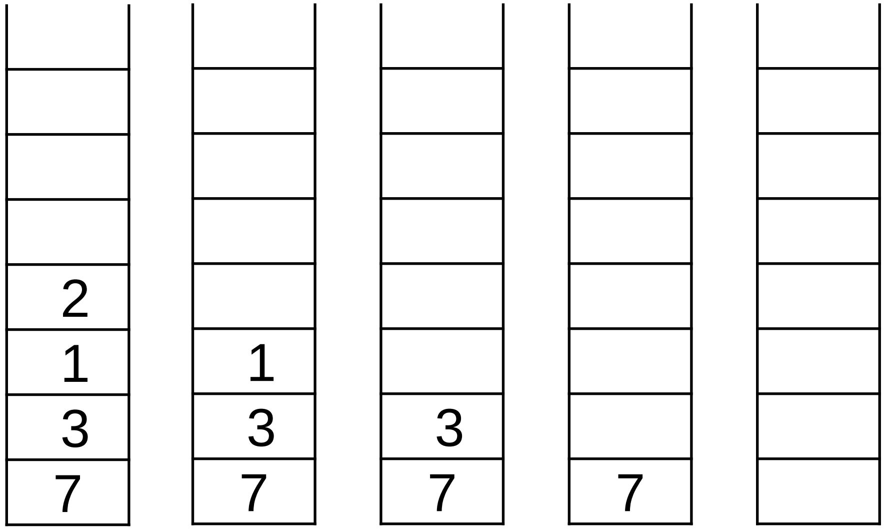


The DFS stack

Depth-first Search (DFS)

- Return to vertex **2** → pop(**4**)
- Return to vertex **1** → pop(**2**)
- Return to vertex **3** → pop(**1**)
- Return to vertex **7** → pop(**3**)
- Return to the invoking method → pop(**7**)





The DFS stack

DFS Properties and Complexity

- Same complexity as BFS
- Same properties with respect to path finding, connected components, and spanning trees
- Edges used to reach unlabeled vertices define a depth-first spanning tree when the graph is connected
- There are problems for which BFS is better than DFS and vice versa

Applications of DFS

- For a weighted graph, DFS traversal of the graph produces the minimum spanning tree and all pair shortest path tree
- Detecting cycle in a graph
- Path Finding
- Topological Sorting
- To test if a graph is bipartite
- Finding Strongly Connected Components of a graph
- Solving puzzles with only one solution

Next Lecture

Applications of Breadth-First Search

Thank you for your attention...

Any question?

Contact:
Department of Information Technology, NITK Surathkal, India
6th Floor, Room: 13
Phone: +91-9477678768
E-mail: shrutilipi@nitk.edu.in, shrutilipi.bhattacharjee@tum.de