

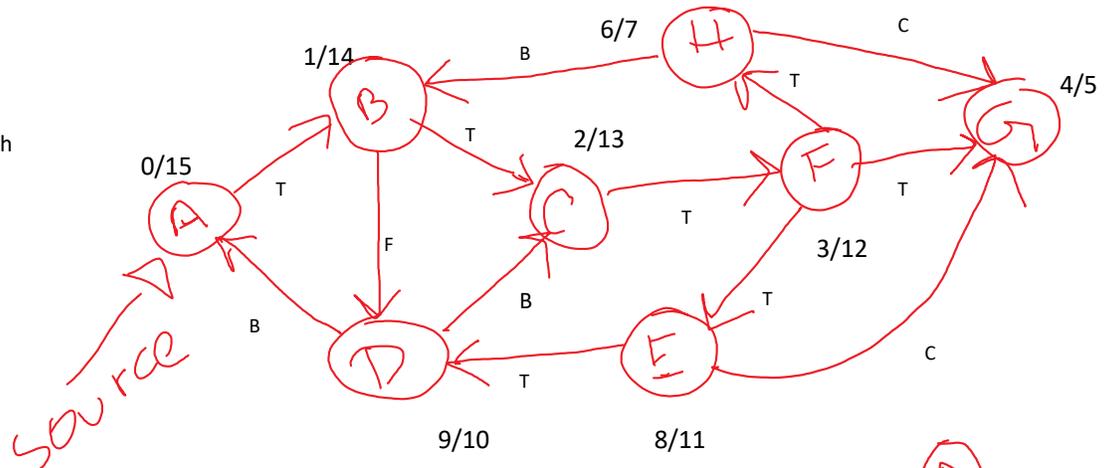
DFS (Depth First Search)

- > Stack is used (LIFO)
- > The Stack is implemented through Recursion
- > A node can be undiscovered, discovered or processed
- > When a node is processed, we backtrack to its source.

Pseudocode:

```

DFS(G,u){
  Visited[u] = true;
  For each v that belongs to Adj[u]{
    If(visited[v] == false){
      DFS(G,v);
    }
  }
}
    
```



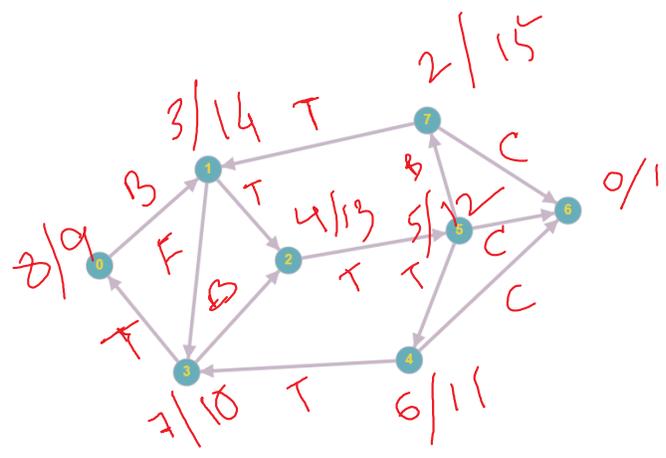
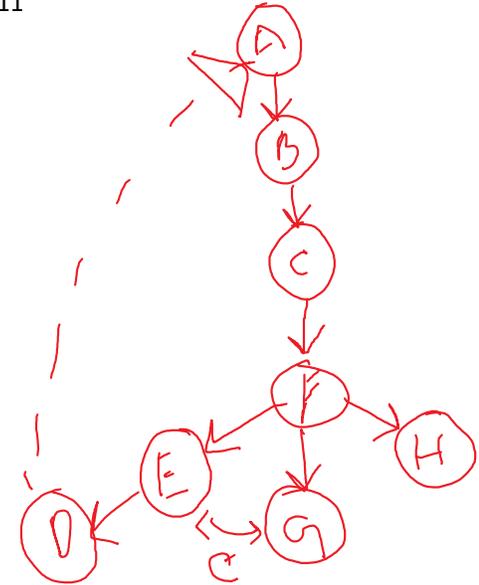
Global time variable = 15

Starting Order: A, B, C, F, G, H, E, D

Finishing Order: A, B, C, F, E, D, H, G

DFS has four types of edges:

- 1) Tree Edge, 2) Forward Edge, 3) Backward Edge, 4) Cross Edge



Source Node is 6.

Starting Order: 6, 7, 1, 2, 5, 4, 3, 0  
 Finishing Order: 7, 1, 2, 5, 4, 3, 0, 6

Time Complexity for BFS & DFS:

- (Matrix Implementation) ->  $O(V^2)$
- (Adj. List Implementation) ->  $O(V+E)$

