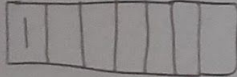


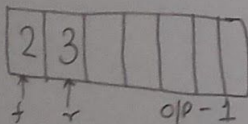
## BFS and DFS

### BFS using Queue

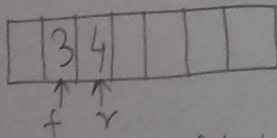
S1: node 1 inserted



S2: node 1 deleted  
adjacent of 1 inserted

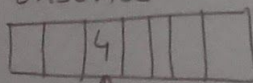


S3: node 2 deleted  
adjacent of 2 inserted



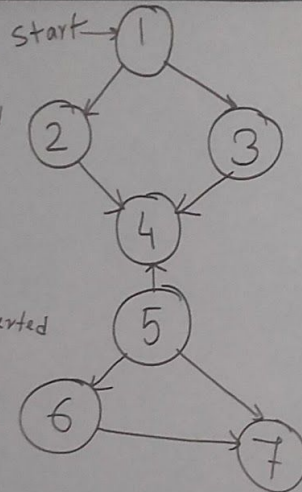
o/p: 1, 2

S4: node 3 deleted  
adjacent of node 3 already  
inserted



o/p: 1, 2, 3

S5: node 4 deleted & Stop  
o/p: 1, 2, 3, 4

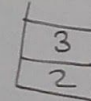


### DFS (using Stack)

S1: node 1 inserted

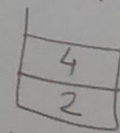


S2: node 1 deleted  
adjacent of node 1 inserted



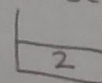
o/p - 1

S3: node 3 deleted  
adjacent of node 3 inserted



o/p: 1, 3

S4: node 4 deleted  
adjacent of node 4 already  
inserted



o/p: 1, 3, 4

S5: node 2 deleted & Stop  
o/p: 1, 3, 4, 2

# BFS using queue

S1: node 1 inserted  
 vis[] → 

0	0	0	0	0	0	0
1	2	3	4	5	6	7

S2: node 1 deleted  
 adjacent of 1 inserted  
 vis[] → 

0	0	0	0	0	0	0
1	2	3	4	5	6	7

S3: node 2 deleted  
 adjacent of 2 inserted  
 vis[] → 

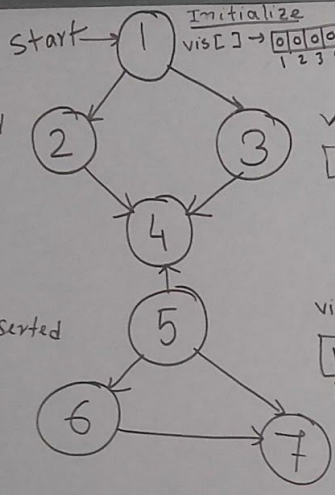
0	0	0	0	0	0	0
1	2	3	4	5	6	7

S4: node 3 deleted  
 adjacent of node 3 already inserted  
 vis[] → 

0	0	0	0	0	0	0
1	2	3	4	5	6	7

S5: node 4 deleted & stop  
 vis[] → 

0	0	0	0	0	0	0
1	2	3	4	5	6	7



# DFS (using Stack)

S1: node 1 inserted  
 vis[] → 

0	0	0	0	0	0	0
1	2	3	4	5	6	7

S2: node 1 deleted  
 adjacent of node 1 inserted  
 vis[] → 

1	0	0	0	0	0	0
1	2	3	4	5	6	7

S3: node 3 deleted  
 adjacent of node 3 inserted  
 vis[] → 

1	1	1	0	0	0	0
1	2	3	4	5	6	7

S4: node 4 deleted  
 adjacent of node 4 already inserted  
 vis[] → 

1	1	1	1	0	0	0
1	2	3	4	5	6	7

S5: node 2 deleted & stop  
 vis[] → 

1	1	1	1	1	0	0
1	2	3	4	5	6	7

## BFS / DFS

### Algorithm

```
void bfs(int s, int n) // void dfs(int s, int n)
{
    int p, i;
    S1 → { add(s); // push(s)
          { vis[s] = 1;

    S2 → { p = delete(); // pop()
          { if(p != 0)
            { print(p);
              while(p != 0) // no. of vertex in graph is O(V)
              {
                S5 → for(i = 1; i <= n; i++) // check connectivity means edges i.e. O(E)
                      { if((a[p][i] != 0) && (vis[i] == 0))
                        { add(i); // push()
                          vis[i] = 1;
                        }
                      }
                p = delete(); // pop()
                if(p != 0)
                { print(p);
                  for(i = 1 to n)
                  { if(vis[i] == 0)
                    { bfs(i, n); // dfs(i)
                    }
                  }
                }
              }
            }
          }
        }
    }
```

after S5 (to visit all nodes in directed graph i.e. nodes 5, 6, 7)