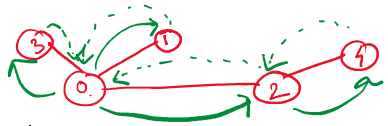
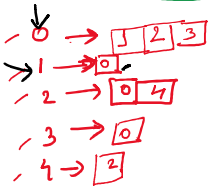
BFS → Visited array, Queue data structure.

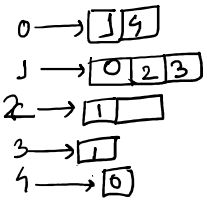
0 2 4 1 3

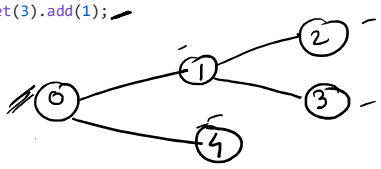




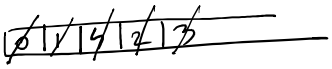
Adjacency list:

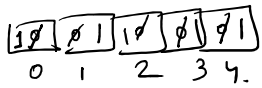
```
ArrayList < ArrayList < Integer >> adj = new ArrayList < > ();
        for (int i = 0; i < 5; i++) {
            adj.add(new ArrayList < > ());
        }
        adj.get(0).add(1);
        adj.get(1).add(0);
        adj.get(0).add(4);
        adj.get(4).add(0);
        adj.get(1).add(2);
        adj.get(2).add(1);
        adj.get(1).add(3);
        adj.get(3).add(1);
```





BFS

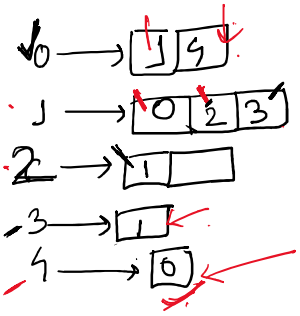
Queue 

Visited 

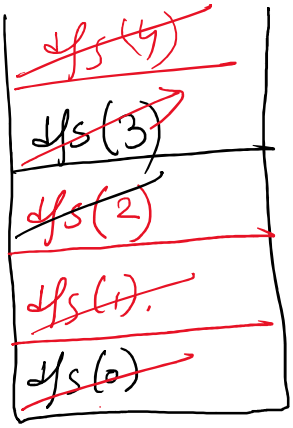
0 1 2 3 4.

print: 0 1 2 3 4

DFS: Recursion:

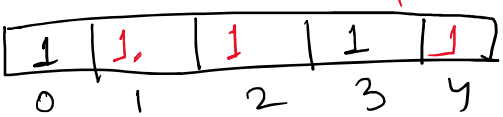


```
dfs(start_node){
    vis[start_node] = 1;
    print(start_node);
    for(int neighbour: adj.get(start_node))
    {
        if(!visited[neighbour]){
            dfs(neighbour);
    } }
```



stack

dfs(0)

vis 

0 1 2 3 4