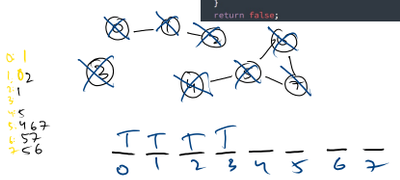
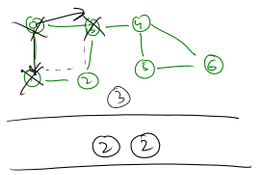
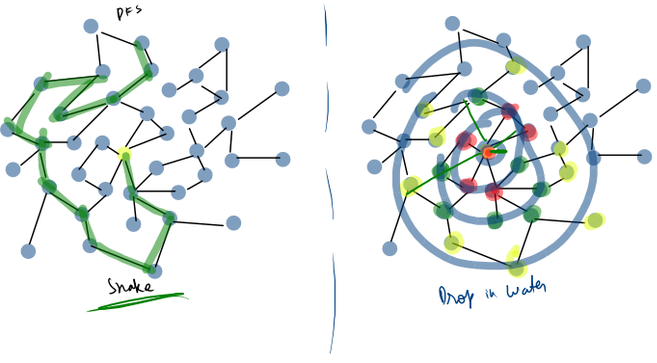
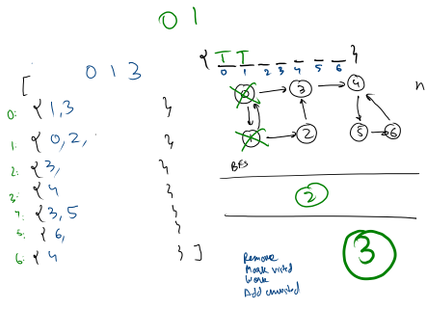
```java
for(int i = 0;i<V;i++){
    if(vis[i]) continue;
    boolean resOfThisComponent = bfs(graph,V,i,vis);
    if(resOfThisComponent) return true;
}
return false;
```





```java
void bfs(  ){
    // your code here
    Queue<Integer> q = new LinkedList<>();
    q.add(x);
    boolean[] visited = new boolean[vertices];
    //start dfs
    while(q.size()>0){
        //remove
        int currNode = q.remove();
        //mark
        if(visited[currNode]) //abstraction / cycle
        visited[currNode] = true;
        //work
        System.out.print(currNode+" ");
        //add unvisited nbrs
        for(int nbr:graph[currNode]){
            if(!visited[nbr]) q.add(nbr);
        }
    }
}
```




```java
void BFS(int x) {
    // your code here
    Queue<Integer> q = new LinkedList<>();
    q.add(X);
    boolean[] visited = new boolean[vertices];
    //start dfs
    while(q.size()>0){
        //remove
        int currNode = q.remove();
        //mark
        if(visited[currNode]) continue; //abstraction / cyc
        visited[currNode] = true;
        //work
        System.out.print(currNode+" ");
        //add unvisited nbrs
        for(int nbr:graph[currNode]){
            if(!visited[nbr]) q.add(nbr);
        }
    }
}
```