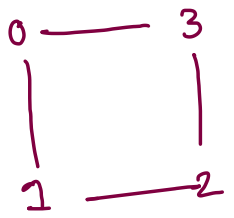
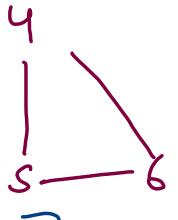
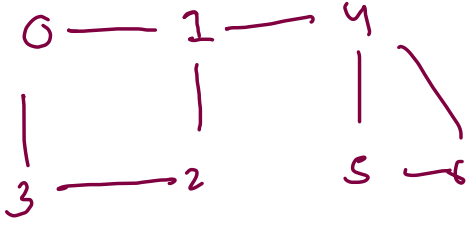
get connected comps

V = 10

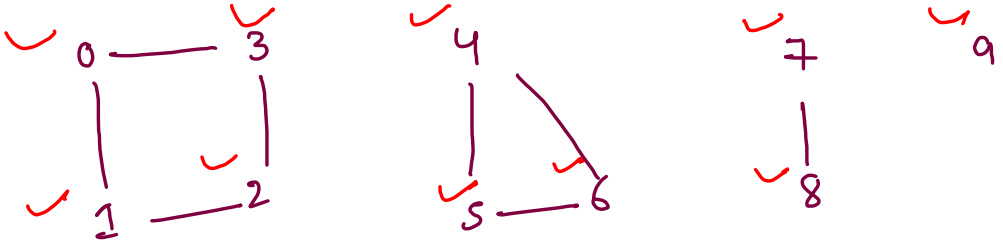


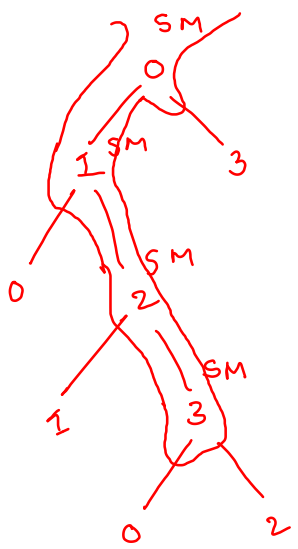


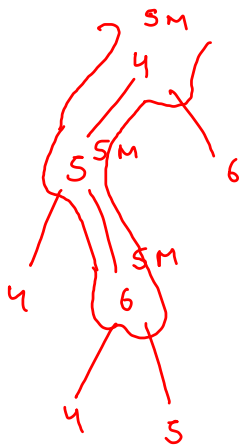
[[0, 1, 2, 3], [4, 5, 6], [7, 8], [9]]

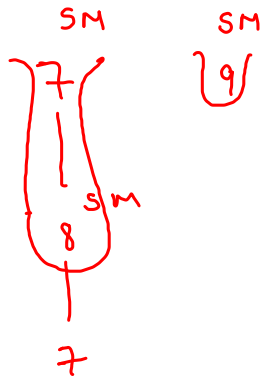


[[0, 1, 2, 3, 4, 5, 6]]









vtx

0 → travel

1 → X

2 → X

3 → X

4 → travel

5 → X

6 → X

7 → travel

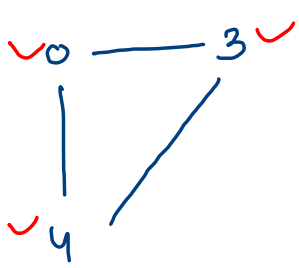
8 → X

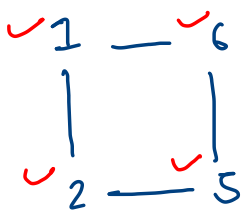
9 → travel

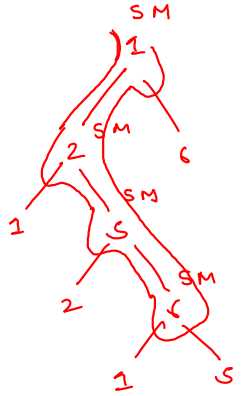
[[0,1,2,3] , [4,5,6]

[7,8], [9]

]







Vtx

- 0 → travel
- 1 → travel
- 2 → X
- 3 → X
- 4 → X
- 5 → X
- 6 → X
- 7 → travel
- 8 → X

travel

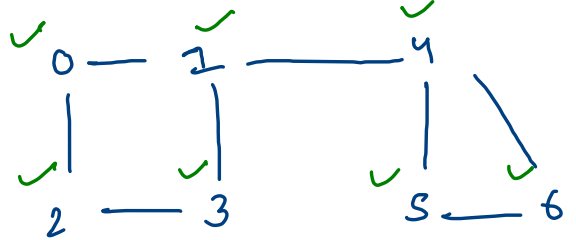
```
list.add(src);

vis[src] = true;

for(Edge edge : graph[src]) {
    int nbr = edge.nbr;

    if(vis[nbr] == false) {
        getSingleComp(graph,nbr,list,vis);
    }
}
```

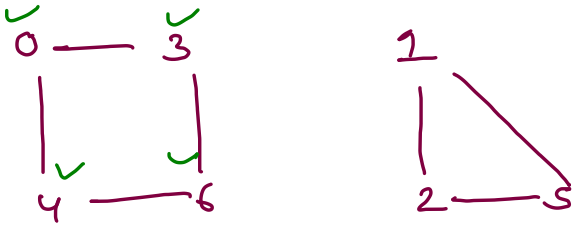
gcc

```
boolean[]vis = new boolean[graph.length];

for(int v = 0 ;v < graph.length;v++) {
    if(vis[v] == false) {
        ArrayList<Integer>scc = new ArrayList<>();
        getSingleComp(graph,v,scc,vis);
        comps.add(scc);
    }
}
```

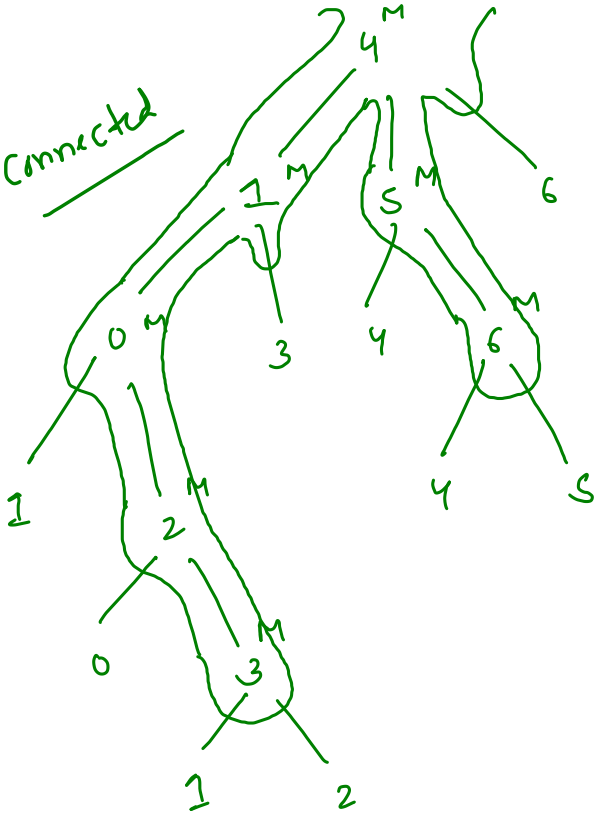
[[0,3,4], [1,2,5,6]

[7, 8]]

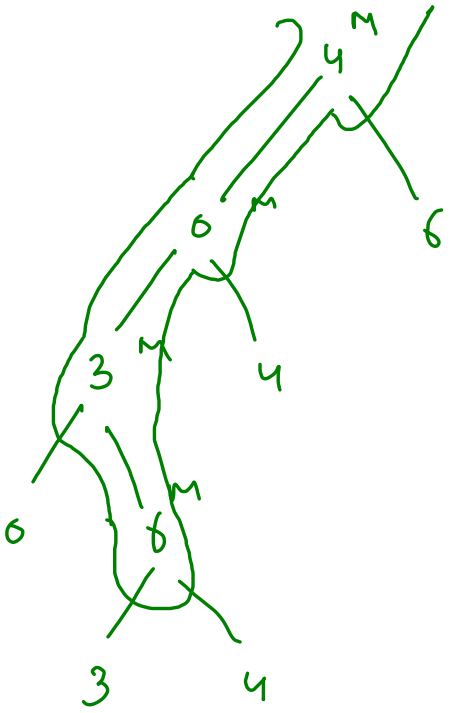


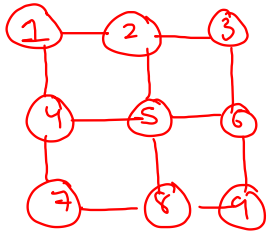


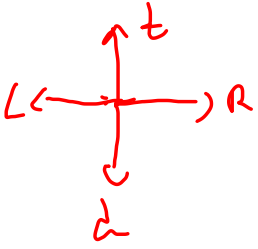
Connected

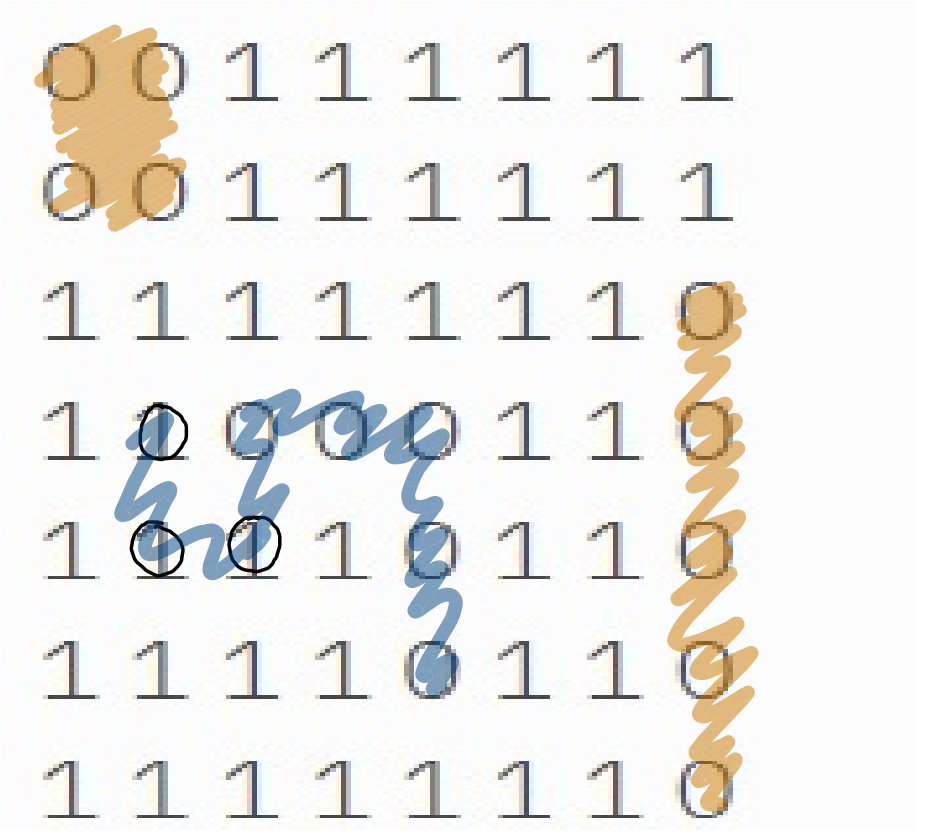


disconnected



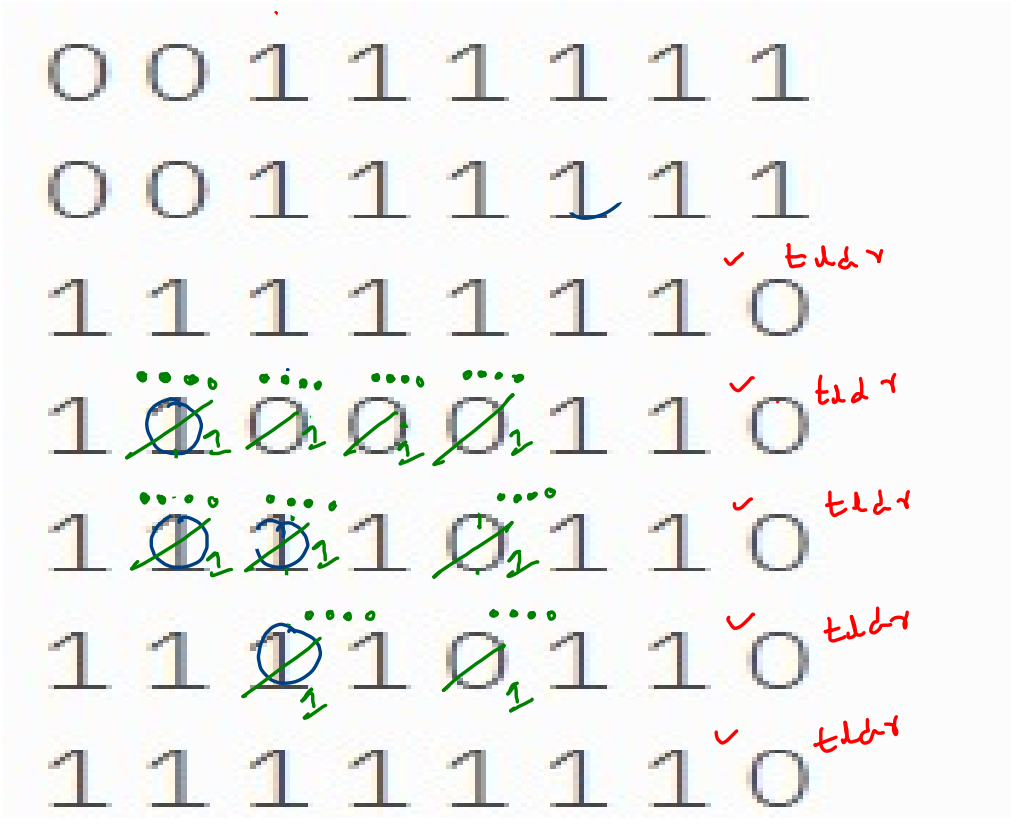






0 -> land

1 -> water




```java
public static int islands(int[][]arr) {

    int count = 0;
    boolean[][]vis = new boolean[arr.length][arr[0].length];

    for(int i=0; i < arr.length;i++) {
        for(int j=0; j < arr[0].length;j++) {
            if(arr[i][j] == 0 && vis[i][j] == false) {
                count++;
                dfs(arr,i,j,vis);
            }
        }
    }

    return count;
```

count = ~~0~~ ~~1~~ ~~2~~ 3

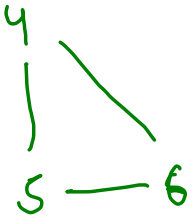
```java
public static void dfs(int[][]arr,int i,int j,boolean[][]vis) {
    if(i < 0 || j < 0 || i == arr.length || j == arr[0].length || vis[i][j] == true || arr[i][j] == 1) {
        return;
    }

    vis[i][j] = true;

    //top nbr
    dfs(arr,i-1,j,vis);

    //Left nbr
    dfs(arr,i,j-1,vis);

    //down nbr
    dfs(arr,i+1,j,vis);

    //right nbr
    dfs(arr,i,j+1,vis);
}

public static int islands(int[][]arr) {
```

arr[i][j] = 1 j

t l d r

1. You are given a number n (representing the number of students). Each student will have an id from 0 to n - 1.
2. You are given a number k (representing the number of clubs)
3. In the next k lines, two numbers are given separated by a space. The numbers are ids of students belonging to same club.
4. You have to find in how many ways can we select a pair of students such that both students are from different clubs.

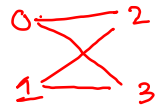
```
7
5
0 1
2 3
4 5
5 6
4 6
```



[2, 2, 3]

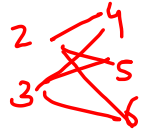
C1 C2 C3

C1 X C2



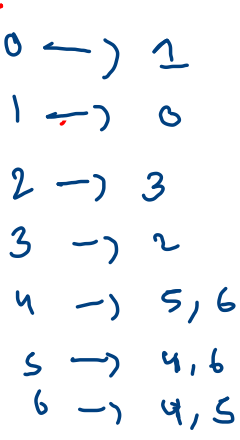
(02, 03, 12, 13)

C2 X C3

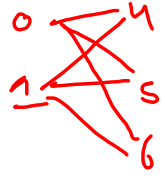


24, 25, 26

34, 35, 36



C1 X C3



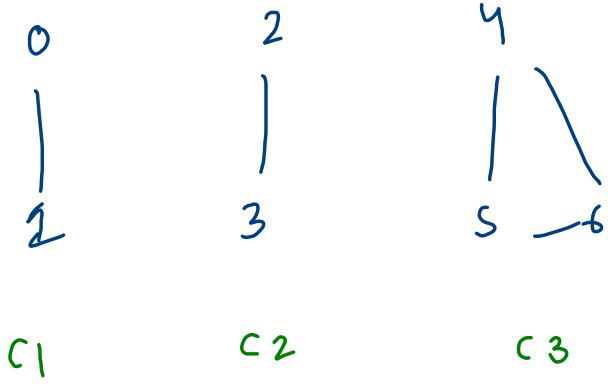
04 14

05 15

06 16

[2, 2, 3]

c1 c2 c3



c1 x c2 = 4

02, 03, 12, 13

c1 x c3 = 6

04, 05, 06, 14, 15, 16

c2 x c3 = 6

24, 25, 26, 34, 35, 36