Shortest Bridge

Sunday, 26 September 2021 7:43 PM

1 → Land
0 → water

No. of islands = 2

Make a bridge from island 1 to island 2 so that the length of bridge is shortest.

	0	1	2	3	4
0	1	1	0	0	1
1	1	1	0	0	1
2	1	1	0	0	0
3	1	1	0	0	0

marking = 1

From any single island, find shortest distance of second land → BFS.

Steps:

① → Find a cell having '1', (If we have all ones from single island, then we can Apply DFS from that one, and add all one in a queue, apply and mark one as well. Barrier are '0', i.e. we can't travel through '0'

BFS → level = ~~1 0 1~~ 2 , First one encounter → level is my ans

BFS)

~~(0,0)~~ | ~~(0,1)~~ | ~~(1,0)~~ | ~~(1,1)~~ | ~~(2,0)~~ | ~~(2,1)~~ | ~~(3,0)~~ | ~~(3,1)~~ | ~~(0,2)~~ | ~~(1,2)~~ | ~~(2,2)~~ | ~~(3,2)~~ | ~~(0,3)~~ | (1,3) | (2,3) | (3,3)

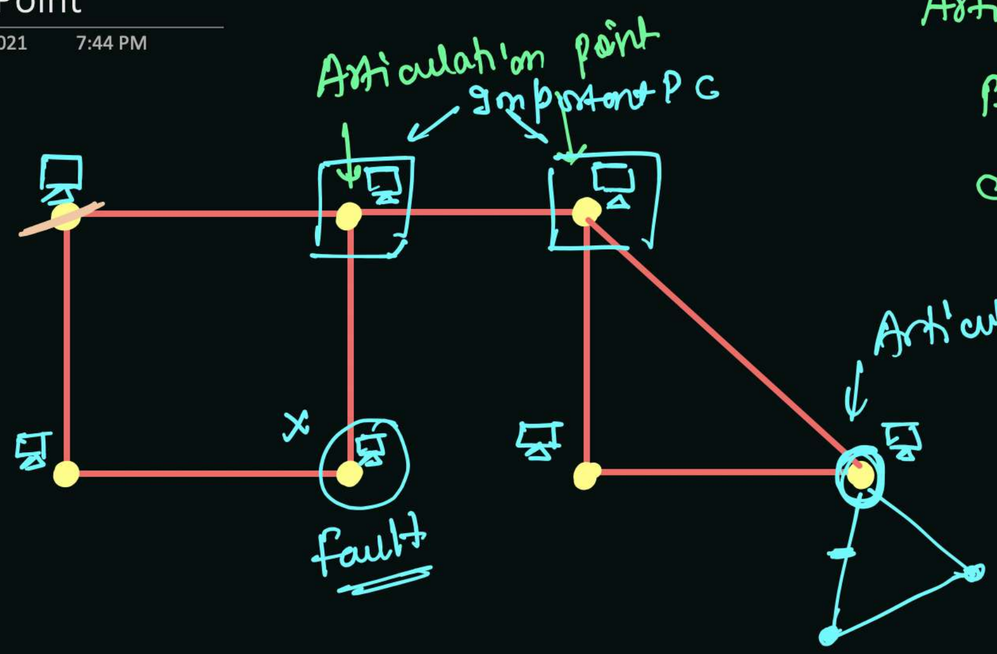
all ones from single island. size = ~~2 4~~ 4

Ans = 2

Articulation Point

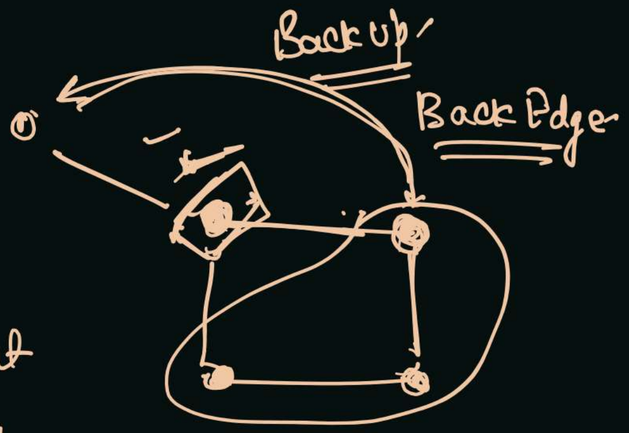
Sunday, 26 September 2021 7:44 PM

Connected Graph.

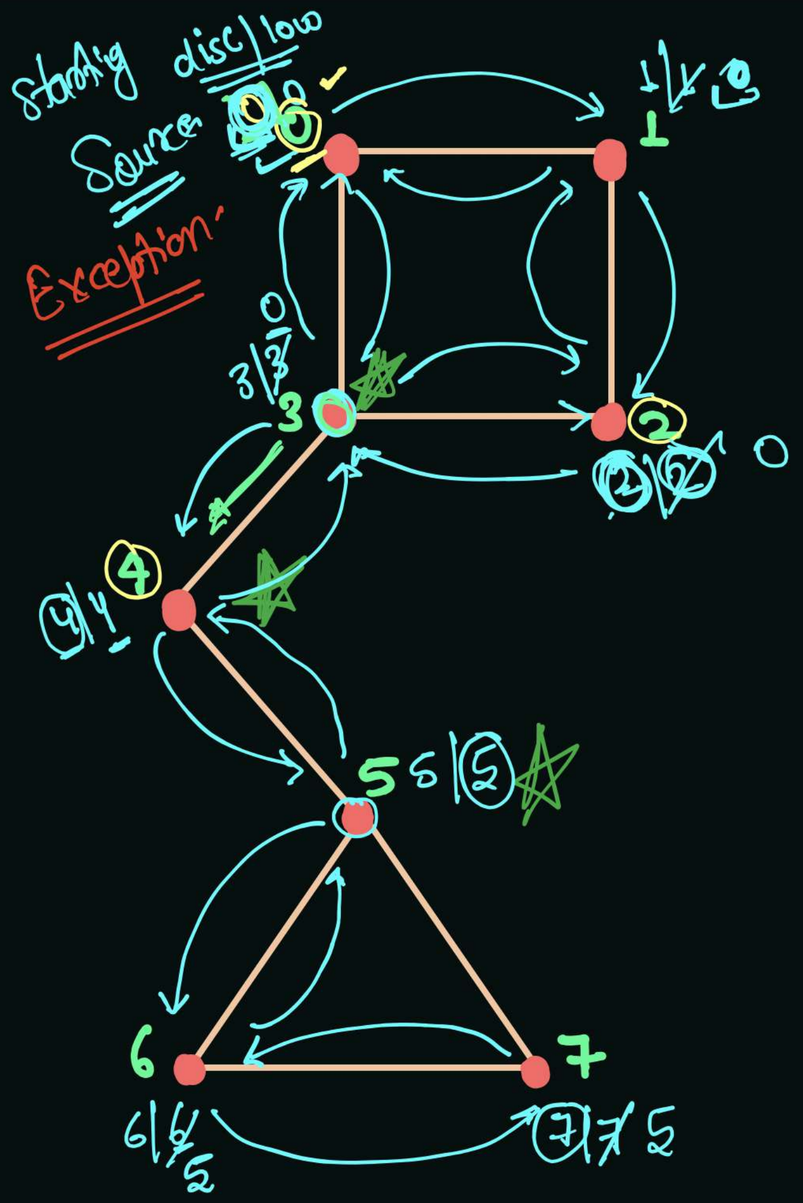


Articulation point → A point from Removal of that point, no. of connected comp. will increase.

Articulation point ??



Brute force ⟶
* Remove vertex one by one and check no of connected component
* If more than one connected comp. is present then that vertex is articulation point



Kihat

- parent []
- discoverytime[] → time of encounter
- low time[] → what is this?

post order

DFS:

disc[src] = low[src] = time;

Loop of neighbour

Stah's int time = 0 1 2

encountered vertex through Edge →

parent

```
if( nbr == par[src]) {
    continue;
}
```

Not Required

Visited neighbour

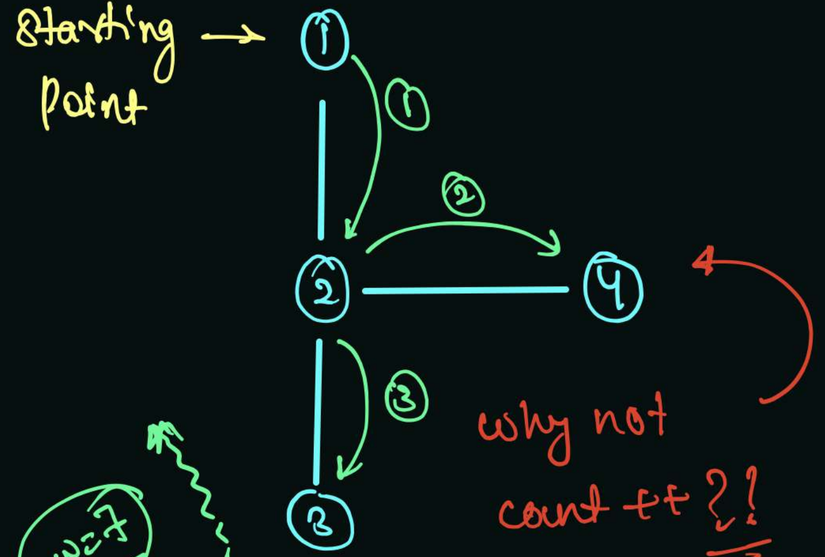
```
else if ( vis[nbr] == true && nbr != par[src]){
    low[src] = Math.min( low[src], disc[nbr]);
```

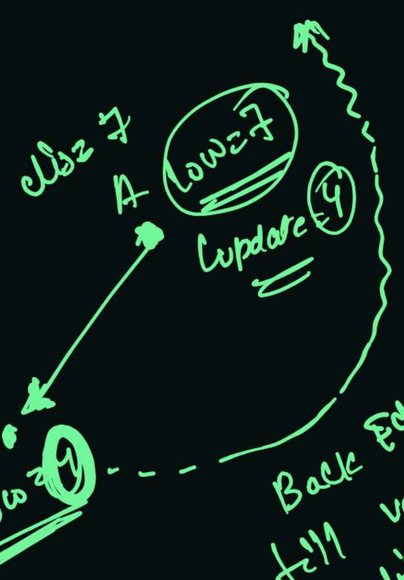
why not low[nbr]??

Unvisited neighbour

```
} else {
    articulation (nbr); → call to DFS.
    low[src] = Math.min ( low[src], low[nbr]);
    if( disc[src] <= low[nbr]) {
        articulation[src] = true;
    }
}
```

Example ①

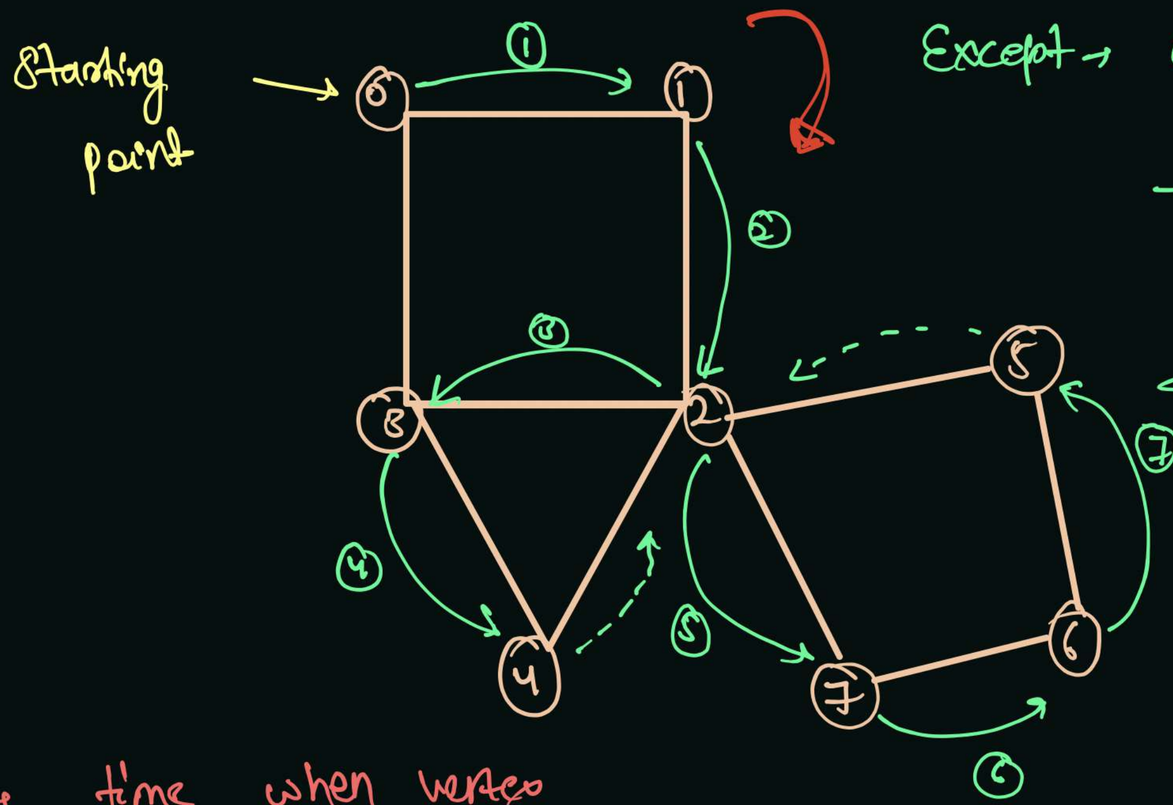




Example 2

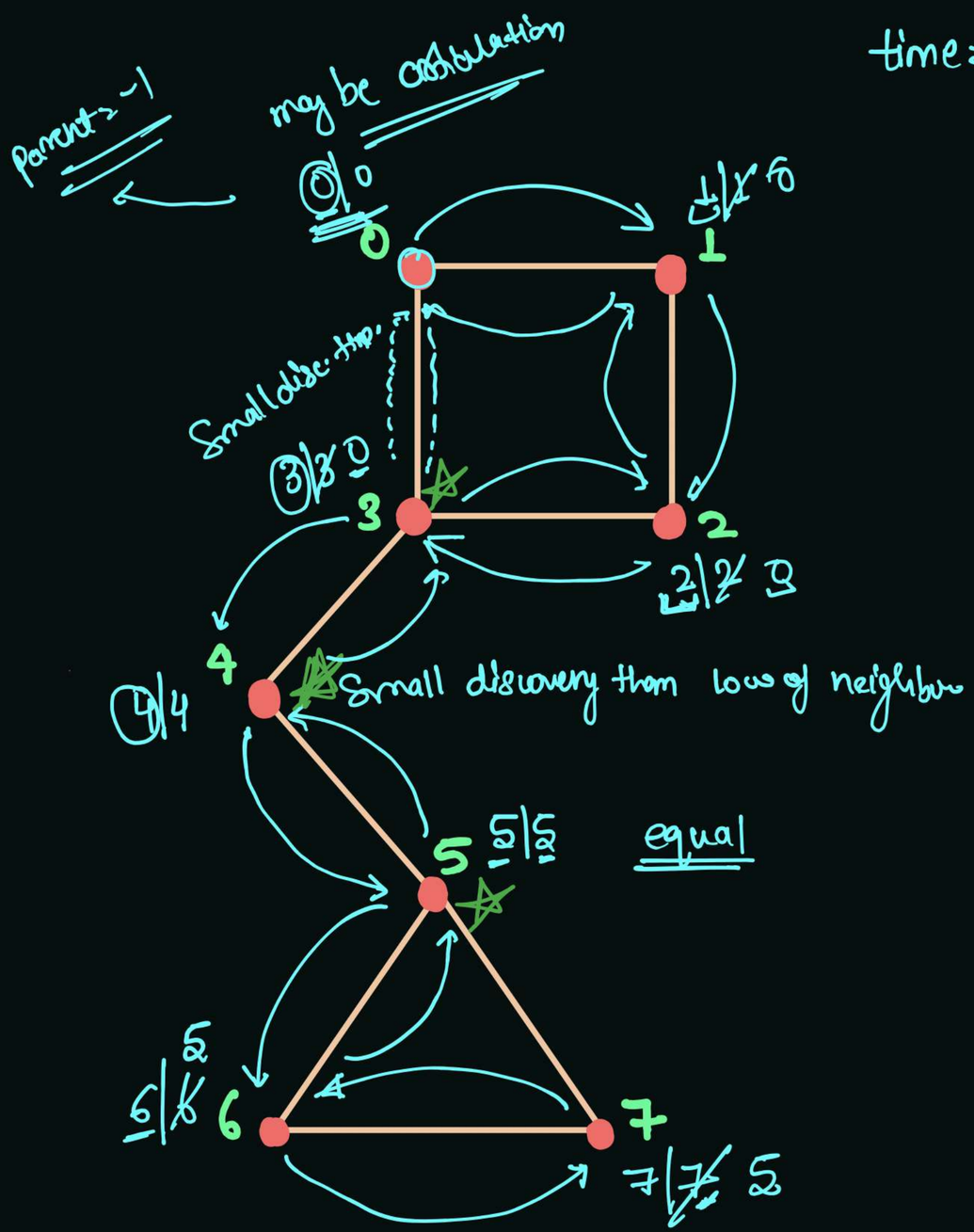
visited nbr →

why not ? → low[src] = min(low[src], low[nbr])



discovery time[] → time when vertex is discover.

low time[] → smallest time of already visited vertex that is reachable from current source




```java
private static void dfsArticulation(ArrayList<ArrayList<Integer>> graph,
int src, boolean[] vis, int[] parent, int[] disc, int[] low, boolean[] arti) {

    disc[src] = low[src] = time;
    time++;
    vis[src] = true;
    for(int nbr : graph.get(src)) {
        if(vis[nbr] == true && parent[src] != nbr) {
            // visited but not parent
            low[src] = Math.min(low[src], disc[nbr]);
        } else if(vis[nbr] == false) {
            // unvisited neighbour
            parent[nbr] = src;
            dfsArticulation(graph, nbr, vis, parent, disc, low, arti);
            low[src] = Math.min(low[src], low[nbr]);
            if(parent[src] == -1) {
                // starting source / root / original source
            } else {
                if(disc[src] <= low[nbr]) {
                    arti[src] = true;
                }
            }
        }
    }
}
```

① why not low[src], low[nbr]

② How to deal with starting src

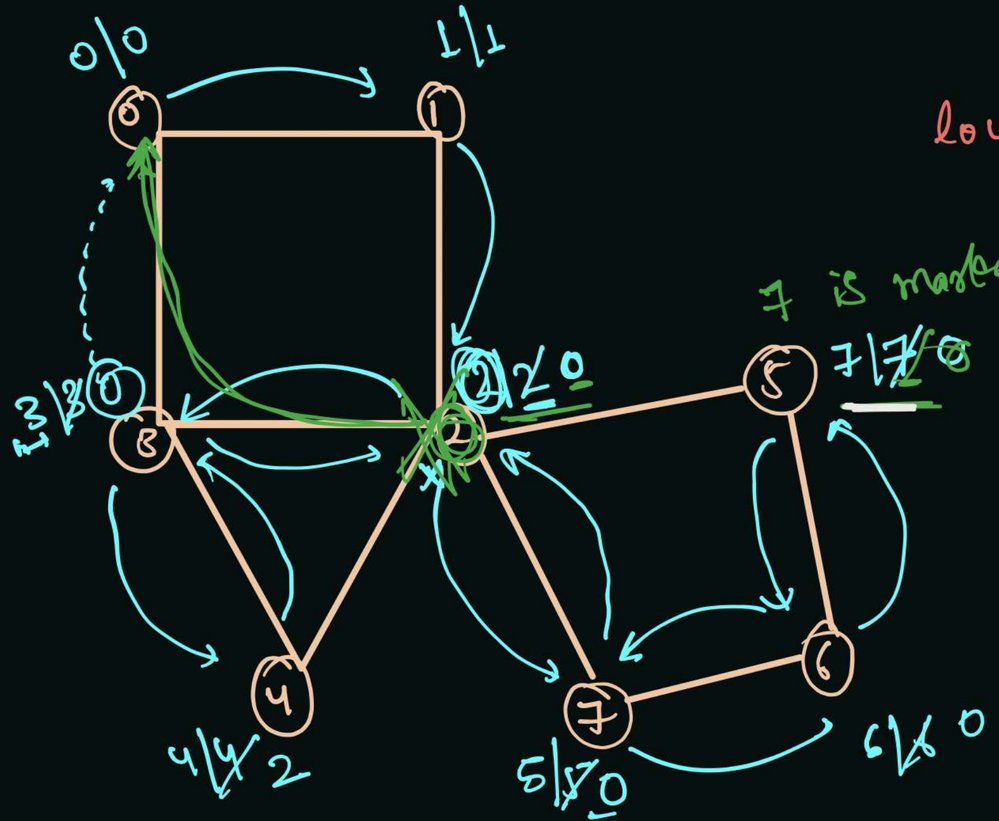
③ why not count++

Not sharp

why's ↓

① why not, low[src] = Math.min(low[src], low[nbr])

when visited but not parent neighbour is encounter.



time = 0 suppose.

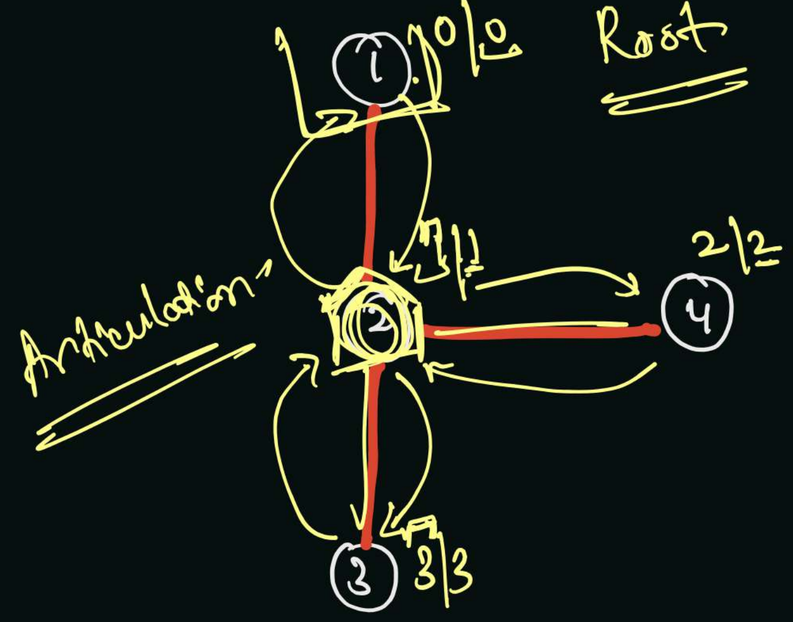
low[src] = Math.min(low[src], low[nbr])

7 is marked its low using low of 2' in already visited neighbour

(disc[src] <= low[nbr])

Articulation

So If we mark with visited neighbour's low point, then that neighbour is required for path, so it can't be an articulation point.

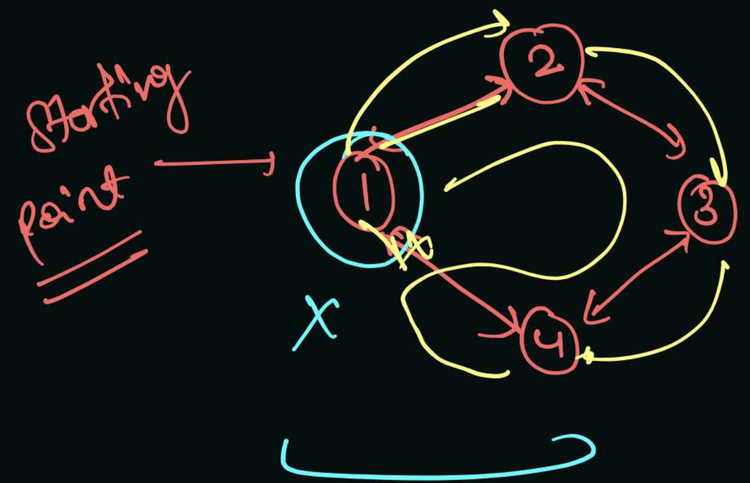


count = 0 < 2

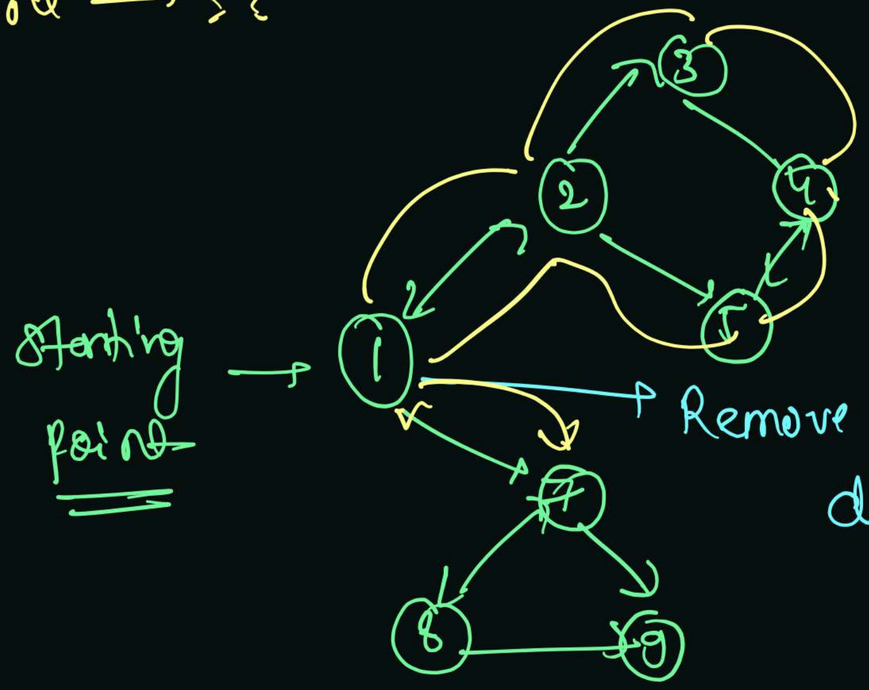
* if vertex if articulation point and it have E edges then we increment count $(e-1)$ time.

⇒ - Count increment will not work to find articulation point, Rather than this, we have to find articulation point using marking in array. or similar technique.

How to manage Starting point → ??



Starting is not articulation point.



← Starting point is articulation point

if. No. of call from starting point is more than 1 time then starting point is articulation point