

924. Minimize Malware Spread

Hard 244 196 Add to List Share

In a network of nodes, each node `i` is directly connected to another node `j` if and only if `graph[i][j] = 1`.

Some nodes `initial` are initially infected by malware. Whenever two nodes are directly connected and at least one of those two nodes is infected by malware, both nodes will be infected by malware. This spread of malware will continue until no more nodes can be infected in this manner.

Suppose `M(initial)` is the final number of nodes infected with malware in the entire network, after the spread of malware stops.

We will remove one node from the initial list. Return the node that if removed, would minimize `M(initial)`. If multiple nodes could be removed to minimize `M(initial)`, return such a node with the smallest index.

Note that if a node was removed from the `initial` list of infected nodes, it may still be infected later as a result of the malware spread.

Example 1:

```
Input: graph = [[1,1,0],[1,1,0],[0,0,1]], initial = [0,1]
Output: 0
```

Example 2:

```
Input: graph = [[1,0,0],[0,1,0],[0,0,1]], initial = [0,2]
Output: 0
```

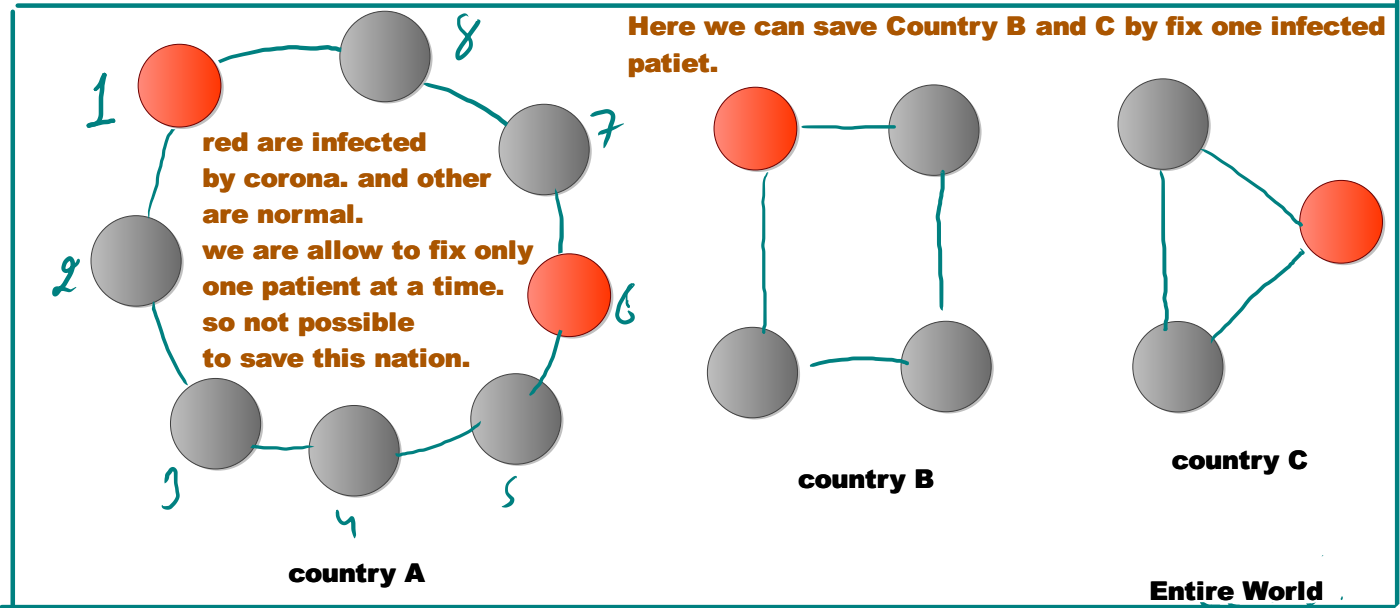
Example 3:

```
Input: graph = [[1,1,1],[1,1,1],[1,1,1]], initial = [1,2]
Output: 1
```

> in this question we have to fix some nodes who are infected during the corona spread. it is certainly possible that node will again get infected by some other infected corona patient.

> here they mention "we will fix one node from the list" so if there are multiples infected corona patient in the country then it is not possible to save the country.

> Firstly, if we save those countries whome have only one infected corona patient then we will sucessfully save the entire nation.
(Its like we stop the spread at intial state and get isolated from world this is the only way to save nation and do more testing in short duration and fix them, if delay in testing then it certainly increase cases)



1. Apply union find so that you get number of components with size of each component (Number of countries in the world with population each country)
2. now iterate over initial infected corona patient and try to find no of patient in each country.
3. those country have only one patient with larger population will be saved first and countries having only one patient with same population then country having smaller index will be saved first.

```
vector<int> par;  
int find(int u)  
{  
    return par[u] == u ? u : par[u] = find(par[u]);  
}
```

```
int minMalwareSpread(vector<vector<int>> &graph, vector<int> &initial)  
{  
    int n = graph.size();  
    for (int i = 0; i < n; i++)  
        par.push_back(i);  
  
    for (int i = 0; i < n; i++)  
        for (int j = 0; j < n; j++)  
            if (i != j && graph[i][j] == 1)  
                par[find(i)] = find(j); //union two set, cycle doesnt matter here so we didn't check just merge two set.  
  
    vector<int> IPC(n, 0); //infected patient count  
    vector<int> POC(n, 0); //population of country  
  
    sort(initial.begin(), initial.end());  
    int ans = initial[0];  
    for (int patient : initial) // set leader will store how many infected patient they have and size of set.  
        IPC[find(patient)]++;  
    for (int i = 0; i < n; i++)  
        POC[find(i)]++;  
  
    int maxPopulation = 0;  
    for (int patient : initial) // iterate over leaders and ask them how many infected patient they have.  
    {  
        if (IPC[par[patient]] == 1 && POC[par[patient]] > maxPopulation)  
        {  
            maxPopulation = POC[par[patient]];  
            ans = patient;  
        }  
    }  
  
    return ans;  
}
```