

2024 -01-20 Handout – Union Find

Q1. The earliest moment when everyone become friends

Link: <https://leetcode.com/problems/the-earliest-moment-when-everyone-become-friends/>

There are n people in a social group labeled from 0 to $n - 1$. You are given an array `logs` where `logs[i] = [timestampi, xi, yi]` indicates that x_i and y_i will be friends at the time `timestampi`.

Friendship is symmetric. That means if a is friends with b , then b is friends with a . Also, person a is acquainted with a person b if a is friends with b , or a is a friend of someone acquainted with b .

Return the earliest time for which every person became acquainted with every other person. If there is no such earliest time, return -1.

Example 1:

Input: `logs =`

```
[[20190101,0,1],[20190104,3,4],[20190107,2,3],  
[20190211,1,5],[20190224,2,4],[20190301,0,3],  
[20190312,1,2],[20190322,4,5]], n = 6
```

Output: 20190301

Example 2:

Input: `logs =`

```
[[0,2,0],[1,0,1],[3,0,3],[4,1,2],[7,3,1]], n  
= 4
```

Output: 3

Q2. Find critical and pseudo critical edges in minimum spanning tree

Link: <https://leetcode.com/problems/find-critical-and-pseudo-critical-edges-in-minimum-spanning-tree>

Given a weighted undirected connected graph with n vertices numbered from 0 to $n - 1$, and an array `edges` where `edges[i] = [ai, bi, weighti]` represents a bidirectional and weighted edge between nodes a_i and b_i . A minimum spanning tree (MST) is a subset of the graph's edges that connects all vertices without cycles and with the minimum possible total edge weight.

Find all the critical and pseudo-critical edges in the given graph's minimum spanning tree (MST). An MST edge whose deletion from the graph would cause the MST weight to increase is called a critical edge. On the other hand, a pseudo-critical edge is that which can appear in some MSTs but not all.

Note that you can return the indices of the edges in any order.

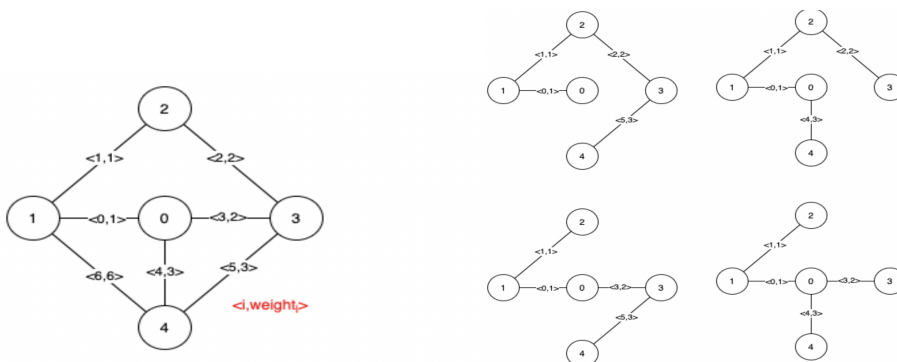
Example 1:

Input: `n = 5, edges = [[0,1,1],[1,2,1],[2,3,2],[0,3,2],[0,4,3],[3,4,3],[1,4,6]]`

Output: `[[0,1],[2,3,4,5]]`

Explanation: The figure above describes the graph.

The following figure shows all the possible MSTs:



Q3. Number of good paths

Link: <https://leetcode.com/problems/number-of-good-paths>

There is a tree (i.e. a connected, undirected graph with no cycles) consisting of n nodes numbered from 0 to $n - 1$ and exactly $n - 1$ edges.

You are given a 0-indexed integer array `vals` of length n where `vals[i]` denotes the value of the i th node. You are also given a 2D integer array `edges` where `edges[i] = [ai, bi]` denotes that there exists an undirected edge connecting nodes ai and bi . A good path is a simple path that satisfies the following conditions:

1. The starting node and the ending node have the same value.
2. All nodes between the starting node and the ending node have values less than or equal to the starting node (i.e. the starting node's value should be the maximum value along the path).

Return the number of distinct good paths.

Note that a path and its reverse are counted as the same path. For example, $0 \rightarrow 1$ is considered to be the same as $1 \rightarrow 0$. A single node is also considered as a valid path.

Example 1:

Input: `vals = [1,3,2,1,3]`, `edges =`

`[[0,1],[0,2],[2,3],[2,4]]`

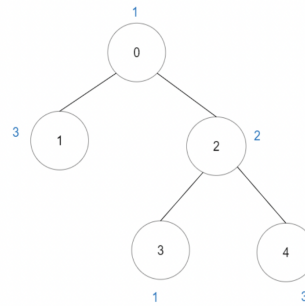
Output: 6

Explanation: There are 5 good paths consisting of a single node.

There is 1 additional good path: $1 \rightarrow 0 \rightarrow 2 \rightarrow 4$.

(The reverse path $4 \rightarrow 2 \rightarrow 0 \rightarrow 1$ is treated as the same as $1 \rightarrow 0 \rightarrow 2 \rightarrow 4$.)

Note that $0 \rightarrow 2 \rightarrow 3$ is not a good path because `vals[2] > vals[0]`.



Q4. Number of Islands II

Link: <https://leetcode.com/problems/number-of-islands-ii/>

You are given an empty 2D binary grid `grid` of size $m \times n$. The grid represents a map where 0's represent water and 1's represent land. Initially, all the cells of grid are water cells (i.e., all the cells are 0's).

We may perform an add land operation which turns the water at position into a land. You are given an array `positions` where `positions[i] = [ri, ci]` is the position (ri, ci) at which we should operate the i th operation.

Return an array of integers `answer` where `answer[i]` is the number of islands after turning the cell (ri, ci) into a land. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Example 1:

Input: `m = 3, n = 3, positions = [[0,0],[0,1],[1,2],[2,1]]`

Output: `[1,1,2,3]`

