2492. Minimum Score of a Path Between Two Cities

My Submissions (/contest/weekly-contest-322/problems/minimum-score-of-a-path-between-two-cities/submissions/)

Back to Contest (/contest/weekly-contest-322/)

User Accepted:	4167
User Tried:	5848
Total Accepted:	4410
Total Submissions:	12054
Difficulty:	Medium

You are given a positive integer `n` representing `n` cities numbered from `1` to `n`. You are also given a **2D** array `roads` where `roads[i] = [`a_i`,` b_i`,` $distance_i$`]` indicates that there is a **bidirectional** road between cities a_i and b_i with a distance equal to $distance_i$. The cities graph is not necessarily connected.

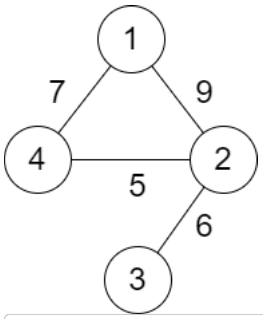
The **score** of a path between two cities is defined as the **minimum** distance of a road in this path.

Return *the **minimum** possible score of a path between cities* `1` *and* `n`.

Note:

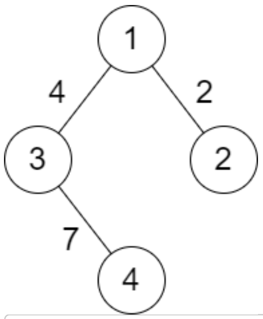
- A path is a sequence of roads between two cities.
- It is allowed for a path to contain the same road **multiple** times, and you can visit cities `1` and `n` multiple times along the path.
- The test cases are generated such that there is **at least** one path between `1` and `n`.

Example 1:



```
Input: n = 4, roads = [[1,2,9],[2,3,6],[2,4,5],[1,4,7]]
Output: 5
Explanation: The path from city 1 to 4 with the minimum score is: 1 -> 2 -> 4. The score of this path is min(9,5) = 5.
It can be shown that no other path has less score.
```

Example 2:



```
Input: n = 4, roads = [[1,2,2],[1,3,4],[3,4,7]]
Output: 2
Explanation: The path from city 1 to 4 with the minimum score is: 1 -> 2 -> 1 -> 3 -> 4. The score of this path is min(2,2,4,7)
```

Constraints:

- `2 <= n <=` 10^5
- `1 <= roads.length <=` 10^5
- `roads[i].length == 3`
- `1 <=` a_i`,` b_i `<= n`
- a_i `!=` b_i
- `1 <=` $distance_i$ `<=` 10^4

- There are no repeated edges.
- There is at least one path between `1` and `n`.

Discuss (https://leetcode.com/problems/minimum-score-of-a-path-between-two-cities/discuss)

JavaScript

```
const initializeGraph = (n) => { let g = []; for (let i = 0; i < n; i++) { g.push([]); } return g; };
const packUG = (g, edges, m) => {
    for (const [u, v, cost] of edges) {
        g[u].push(v);
        g[v].push(u);
        m.set(u + " " + v, cost)
        m.set(v + " " + u, cost)
    }
};

function DJSet(n, m) {
    let parent = Array(n).fill(-1), min = Array(n).fill(Number.MAX_SAFE_INTEGER);
    return { find, union, count, equiv, par }
    function find(x) {
        return parent[x] < 0 ? x : parent[x] = find(parent[x]);
    }
    function union(x, y) {
        let ke = x + " " + y, cost = m.get(ke);
        min[x] = Math.min(min[x], cost);
        min[y] = Math.min(min[y], cost);
        x = find(x);
        y = find(y);
        if (x != y) {
            if (parent[x] < parent[y]) [x, y] = [y, x];
            parent[x] += parent[y];
            parent[y] = x;

        }
        return x == y;
    }
    function count() {
        let res = Number.MAX_SAFE_INTEGER;
        for (let i = 1; i < n; i++) {
            if (equiv(1, i)) res = Math.min(res, min[i]);
        }
        return res;
    }
    function equiv(x, y) { // isConnected
        return find(x) == find(y);
    }
    function par() {
        return parent;
    }
}

const minScore = (n, roads) => {
    let g = initializeGraph(n + 1), m = new Map();
    packUG(g, roads, m);
    let ds = new DJSet(n + 1, m);
    for (const [u, v, cost] of roads) ds.union(u, v)
    return ds.count();
};
```

Custom Testcase   Use Example Testcases

Run   Submit

Submission Result: Accepted (/submissions/detail/854265689/)   More Details (/submissions/detail/854265689/)

Share your acceptance!

3