6255. 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/)

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.

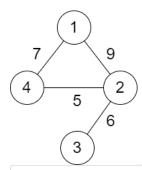
Return the $\mbox{\it minimum}$ possible score of a path between cities $\,1\,$ and $\,n$.

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 $\, n \,$.

User Accepted:	5
User Tried:	11
Total Accepted:	5
Total Submissions:	14
Difficulty:	Medium

Example 1:

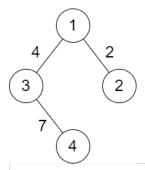


Input: n = 4, roads = [[1,2,9],[2,3,6],[2,4,5],[1,4,7]]

Output: 5

Explanation: The path from city 1 to 4 with the minimum score is: $1 \rightarrow 2 \rightarrow 4$. 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]]

Explanation: The path from city 1 to 4 with the minimum score is: $1 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 4$. 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
- $1 \le a_i$, $b_i \le n$
- $a_i != b_i$
- 1 <= distance $_i$ <= 10^4

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

```
C
 JavaScript
                                                                                                                                    ď
      const initializeGraph = (n) => { let g = []; for (let i = 0; i < n; i++) { g.push([]); } return g; };
      const packUG = (g, edges, m) \Rightarrow \{
  2 🔻
  З ч
          for (const [u, v, cost] of edges) {
  4
               g[u].push(v);
  5
               g[v].push(u);
m.set(u + " " + v, cost)
  6
  7
               m.set(v + " " + u, cost)
  8
          }
  9
      };
 10
 11 ▼
      const minScore = (n, roads) => {
 12
          let g = initializeGraph(n + 1), m = new Map(), q = [1], vis = Array(n + 1).fill(false), res = new Map()
      Number.MAX_SAFE_INTEGER;
 13
          packUG(g, roads, m);
 14
          vis[1] = true;
 15 ▼
          while (q.length) {
               let cur = q.shift();
 16
               for (const child of g[cur]) {
   let cost = m.get(cur + " " + child);
 17 ▼
 18
                    res = Math.min(res, cost);
 19
 20 •
                    if (!vis[child]) {
                        q.push(child);
 21
 22
                        vis[child] = true;
 23
 24
               }
 25
          }
 26
          return res;
 27
      };
☐ Custom Testcase
                       Use Example Testcases
                                                                                                                                Run
                                                                                                                                          △ Submit
Submission Result: Accepted (/submissions/detail/854253409/) ?
                                                                               More Details > (/submissions/detail/854253409/)
Share your acceptance!
              ∢3
Copyright © 2022 LeetCode
Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)
United States (/region)
```