# 5632. Checking Existence of Edge Length Limited Paths

My Submissions (/contest/weekly-contest-220/problems/checking-existence-of-edge-length-limited-paths/submissions/)

Back to Contest (/contest/weekly-contest-220/)

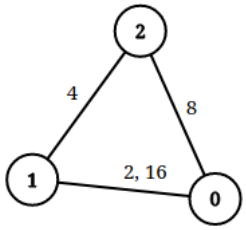
An undirected graph of `n` nodes is defined by `edgeList`, where `edgeList[i] = [u_i, v_i, dis_i]` denotes an edge between nodes `u_i` and `v_i` with distance `dis_i`. Note that there may be **multiple** edges between two nodes.

Given an array `queries`, where `queries[j] = [p_j, q_j, limit_j]`, your task is to determine for each `queries[j]` whether there is a path between `p_j` and `q_j` such that each edge on the path has a distance **strictly less than** `limit_j`.

Return *a* ***boolean array*** `answer`, *where* `answer.length == queries.length` *and the* `j^th` *value of* `answer` *is* `true` *if there is a path for* `queries[j]` *is* `true`, *and* `false` *otherwise.*

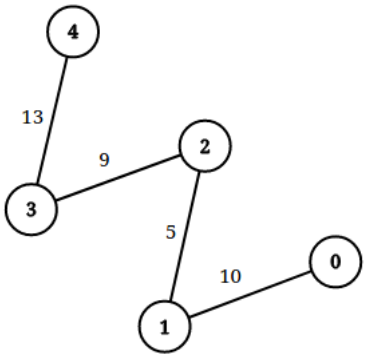
| | |
|---|---|
| User Accepted: | 0 |
| User Tried: | 0 |
| Total Accepted: | 0 |
| Total Submissions: | 0 |
| Difficulty: | Hard |

**Example 1:**




```
Input: n = 3, edgeList = [[0,1,2],[1,2,4],[2,0,8],[1,0,16]], queries = [[0,1,2],[0,2,5]]
Output: [false,true]
Explanation: The above figure shows the given graph. Note that there are two overlapping edges between 0 and 1 with distances 2
For the first query, between 0 and 1 there is no path where each distance is less than 2, thus we return false for this query.
For the second query, there is a path (0 -> 1 -> 2) of two edges with distances less than 5, thus we return true for this query
```

**Example 2:**




```
Input: n = 5, edgeList = [[0,1,10],[1,2,5],[2,3,9],[3,4,13]], queries = [[0,4,14],[1,4,13]]
Output: [true,false]
Exaplanation: The above figure shows the given graph.
```

**Constraints:**

- `2 <= n <= 10^5`
- `1 <= edgeList.length, queries.length <= 10^5`
- `edgeList[i].length == 3`
- `queries[j].length == 3`

- $0 <= u_i, v_i, p_j, q_j <= n - 1$
- $u_i != v_i$
- $p_j != q_j$
- $1 <= dis_i, limit_j <= 10^9$
- There may be **multiple** edges between two nodes.

TypeScript

```
function distanceLimitedPathsExist(n: number, edgeList: number[][], queries: number[][]): boolean[] {

};
```

**Custom Testcase** Use Example Testcases

Run Submit