

## 5921. Maximum Path Quality of a Graph

[My Submissions \(/contest/weekly-contest-266/problems/maximum-path-quality-of-a-graph/submissions/\)](#)
[Back to Contest \(/contest/weekly-contest-266/\)](#)

There is an **undirected** graph with  $n$  nodes numbered from  $0$  to  $n - 1$  (**inclusive**). You are given a **0-indexed** integer array `values` where `values[i]` is the **value** of the  $i^{\text{th}}$  node. You are also given a **0-indexed** 2D integer array `edges`, where each `edges[j] = [uj, vj, timej]` indicates that there is an undirected edge between the nodes  $u_j$  and  $v_j$ , and it takes `timej` seconds to travel between the two nodes. Finally, you are given an integer `maxTime`.

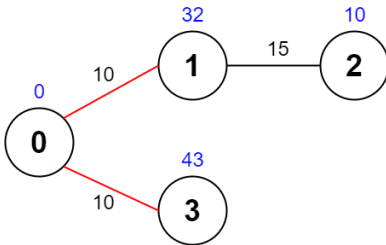
A **valid path** in the graph is any path that starts at node  $0$ , ends at node  $0$ , and takes **at most** `maxTime` seconds to complete. You may visit the same node multiple times. The **quality** of a valid path is the **sum** of the values of the **unique nodes** visited in the path (each node's value is added **at most once** to the sum).

Return the **maximum** quality of a valid path.

**Note:** There are **at most four** edges connected to each node.

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

**Example 1:**



**Input:** `values = [0,32,10,43]`, `edges = [[0,1,10],[1,2,15],[0,3,10]]`, `maxTime = 49`

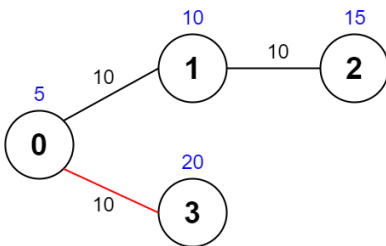
**Output:** 75

**Explanation:**

One possible path is  $0 \rightarrow 1 \rightarrow 0 \rightarrow 3 \rightarrow 0$ . The total time taken is  $10 + 10 + 10 + 10 = 40 \leq 49$ .

The nodes visited are 0, 1, and 3, giving a maximal path quality of  $0 + 32 + 43 = 75$ .

**Example 2:**



**Input:** `values = [5,10,15,20]`, `edges = [[0,1,10],[1,2,10],[0,3,10]]`, `maxTime = 30`

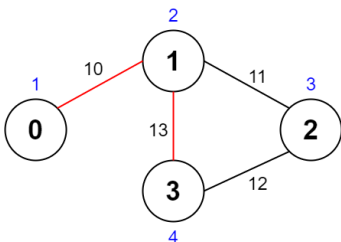
**Output:** 25

**Explanation:**

One possible path is  $0 \rightarrow 3 \rightarrow 0$ . The total time taken is  $10 + 10 = 20 \leq 30$ .

The nodes visited are 0 and 3, giving a maximal path quality of  $5 + 20 = 25$ .

**Example 3:**



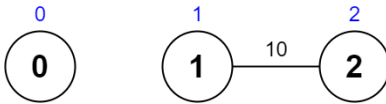
**Input:** values = [1,2,3,4], edges = [[0,1,10],[1,2,11],[2,3,12],[1,3,13]], maxTime = 50

**Output:** 7

**Explanation:**

One possible path is 0 → 1 → 3 → 1 → 0. The total time taken is 10 + 13 + 13 + 10 = 46 ≤ 50.  
The nodes visited are 0, 1, and 3, giving a maximal path quality of 1 + 2 + 4 = 7.

**Example 4:**



**Input:** values = [0,1,2], edges = [[1,2,10]], maxTime = 10

**Output:** 0

**Explanation:**

The only path is 0. The total time taken is 0.

The only node visited is 0, giving a maximal path quality of 0.

**Constraints:**

- $n == \text{values.length}$
- $1 \leq n \leq 1000$
- $0 \leq \text{values}[i] \leq 10^8$
- $0 \leq \text{edges.length} \leq 2000$
- $\text{edges}[j].\text{length} == 3$
- $0 \leq u_j < v_j \leq n - 1$
- $10 \leq \text{time}_j, \text{maxTime} \leq 100$
- All the pairs  $[u_j, v_j]$  are **unique**.
- There are **at most four** edges connected to each node.
- The graph may not be connected.

JavaScript



```

1  const initializeGraph = (n) => { let G = []; for (let i = 0; i < n; i++) { G.push([]); } return G; };
2  const packUGCost = (G, Edges) => { for (const [u, v, cost] of Edges) { G[u].push([v, cost]); G[v].push([u, cost]); } };
3
4  let v, g, res, depth;
5  const maximalPathQuality = (values, edges, maxTime) => {
6    let n = values.length;
7    v = values, g = initializeGraph(n), res = 0, depth = Array(n).fill(0);
8    packUGCost(g, edges);
9    // pr(g);
10   dfs(0, maxTime, 0);
11   return res;
12 };
13
14 const dfs = (cur, time, sum) => {
15   // pr(cur, time, sum, depth);
16   if (time < 0) return;
17   depth[cur]++;
18   if (depth[cur] == 1) sum += v[cur]; // depth is 1, come back to starting node 0
19   if (cur == 0) res = Math.max(sum, res);
20   for (const [child, cost] of g[cur]) dfs(child, time - cost, sum)
21   depth[cur]--; // backtracking, depth -1
22 };
  
```

☐ Custom Testcase

Use Example Testcases

Run


Submit

**Submission Result: Accepted** (/submissions/detail/583285472/) ?

More Details &gt; (/submissions/detail/583285472/)

Share your acceptance!

---

Copyright © 2021 LeetCode[Help Center \(/support\)](/support/) | [Jobs \(/jobs\)](/jobs/) | [Bug Bounty \(/bugbounty\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student\)](/student/) | [Terms \(/terms\)](/terms/) | [Privacy Policy \(/privacy\)](/privacy/) [United States \(/region\)](/region/)