5861. Maximum Earnings From Taxi

My Submissions (/contest/biweekly-contest-61/problems/maximum-earnings-from-taxi/submissions/)

Back to Contest (/contest/biweekly-contest-61/)

There are `n` points on a road you are driving your taxi on. The `n` points on the road are labeled from `1` to `n` in the direction you are going, and you want to drive from point `1` to point `n` to make money by picking up passengers. You cannot change the direction of the taxi.

The passengers are represented by a **0-indexed** 2D integer array `rides`, where `rides[i] = [start_i, end_i, tip_i]` denotes the i^{th} passenger requesting a ride from point `start_i` to point `end_i` who is willing to give a `tip_i` dollar tip.

For **each** passenger `i` you pick up, you **earn** `end_i - start_i + tip_i` dollars. You may only drive **at most one** passenger at a time.

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

Given `n` and `rides`, return *the **maximum** number of dollars you can earn by picking up the passengers optimally.*

Note: You may drop off a passenger and pick up a different passenger at the same point.

Example 1:

```
Input: n = 5, rides = [[2,5,4],[1,5,1]]
Output: 7
Explanation: We can pick up passenger 0 to earn 5 - 2 + 4 = 7 dollars.
```

Example 2:

```
Input: n = 20, rides = [[1,6,1],[3,10,2],[10,12,3],[11,12,2],[12,15,2],[13,18,1]]
Output: 20
Explanation: We will pick up the following passengers:
- Drive passenger 1 from point 3 to point 10 for a profit of 10 - 3 + 2 = 9 dollars.
- Drive passenger 2 from point 10 to point 12 for a profit of 12 - 10 + 3 = 5 dollars.
- Drive passenger 5 from point 13 to point 18 for a profit of 18 - 13 + 1 = 6 dollars.
We earn 9 + 5 + 6 = 20 dollars in total.
```

Constraints:

- `1 <= n <= 10^5`
- `1 <= rides.length <= 3 * 10^4`
- `rides[i].length == 3`
- `1 <= start_i < end_i <= n`
- `1 <= tip_i <= 10^5`

JavaScript

```
const initializeGraph = (n) => { let G = []; for (let i = 0; i < n; i++) { G.push([]); } return G; };

const maxTaxiEarnings = (n, rides) => {
    let g = initializeGraph(n);
    for (const [u, v, cost] of rides) g[v - 1].push([u - 1, v - u + cost]);
    let dp = Array(n).fill(0);
    for (let i = 1; i < n; i++) {
        dp[i] = Math.max(dp[i], dp[i - 1]);
        for (const [next_node, curCost] of g[i]) {
            dp[i] = Math.max(dp[i], dp[next_node] + curCost);
```

```
        }
    }
    return dp[n - 1];
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/557038940/) | More Details > (/submissions/detail/557038940/)

Share your acceptance!