Problem 2008: Maximum Earnings From Taxi

Problem Information

Difficulty: Medium
Acceptance Rate: 45.88%
Paid Only: No
Tags: Array, Hash Table, Binary Search, Dynamic Programming, Sorting

Problem Description

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] = [starti, endi, tipi]` denotes the `ith` passenger requesting a ride from point `starti` to point `endi` who is willing to give a `tipi` dollar tip.

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

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 <= 105` * `1 <= rides.length <= 3 * 104` * `rides[i].length == 3` * `1 <= starti < endi <= n` * `1 <= tipi <= 105`

Code Snippets

C++:

```
class Solution {
public:
long long maxTaxiEarnings(int n, vector<vector<int>>& rides) {

}
};
```

Java:

```
class Solution {
public long maxTaxiEarnings(int n, int[][] rides) {

}
}
```

Python3:

```
class Solution:
def maxTaxiEarnings(self, n: int, rides: List[List[int]]) -> int:
```