Problems

Courses Problem Editorial Submissions Comments

Max Coins

Job Fair Live

185

Medium Accuracy: **52.73%** Submissions: **23K+** Points: **4**

Contests

POTD

You are given a 2D integer array **ranges** whose length is **n** where **ranges[i]=[start$_i$, end,coins$_i$]** means all integers from start$_i$ to end$_i$ inclusive start$_i$ and end$_i$ are present and we get **coins$_i$** amount of **coins** when we select this **ith** range. You can select **at most** two intervals so as to collect maximum coins but if you select two ranges then those two ranges **should not intersect or overlap but can touch each other.**

Note: You can select at max 2 ranges and they should not intersect with each other but they can touch themselves.

Example 1:

Input :
n=3
ranges={{1,3,4},{2,3,5},{3,4,2}}
Output: 7
Explanation:
We can see that we can take 2nd and 3rd ranges as they are not intersecting (only touching) we get maximum Coins by taking these ranges(5+2=7).

Example 2:

Input :
n=5
ranges={{1,3,4},{2,3,5},{3,4,2},{5,8,9},{2,8,10}}
Output: 14
Explanation:
We can see that we can take 2nd and 4th ranges as they are not intersecting

Menu

we get maximum Coins(5+9=14) by taking these ranges.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maxCoins()** which takes an integer **n**(**length of ranges**), integer 2D integer array **ranges,** and you have to return the **maximum** number of **coins** you got after selecting **at most** two ranges that are not intersecting.

Expected Time Complexity: O(nlogn)

Expected Space Complexity: O(n)

Constraints:

$1 <= \mathbf{n} <= 10^5$

$0 <= \mathbf{ranges}[i][0] <= \mathbf{ranges}[i][1] <= 10^9$

$0 <= \mathbf{ranges}[i][2](coins) <= 10^6$

The sum of n over all test cases won't exceed 10^6

View Bookmarked Problems

Topic Tags

C++ (g++ 5.4) Start Timer

```
1  // } Driver Code Ends
9  //User function Template for C++
10
11 class Solution{
12 public:
13     int maxCoins(int n,vector<vector<int>> &ranges){
14         // Code here
15         sort(ranges.begin(), ranges.end(), [&](vector<int> &a, vector<int> &b){
16             if(a[0] == b[0])
17                 return a[1] < b[1];
18             return a[0] < b[0];
19         });
20
21         vector<int> post(n);
22         post[n - 1] = ranges[n - 1][2];
23
24         for(int i = n - 2; i > -1; i--)
25             post[i] = max(post[i + 1], ranges[i][2]);
26
27         int ans = 0;
28
29         for(int i = 0; i < n; i++){
30             int l = i;
```

Menu

```
34                  int mid = l + (h - l) / 2;
35
36                  if(ranges[mid][0] >= ranges[i][1])
37                      h = mid;
38                  else
39                      l = mid;
40              }
41
42              int cur = 0;
43              if(h < n)
44                  cur = post[h];
45
46              ans = max(ans, ranges[i][2] + cur);
47          }
48
49          return ans;
50      }
51  };
52  // } Driver Code Ends
```