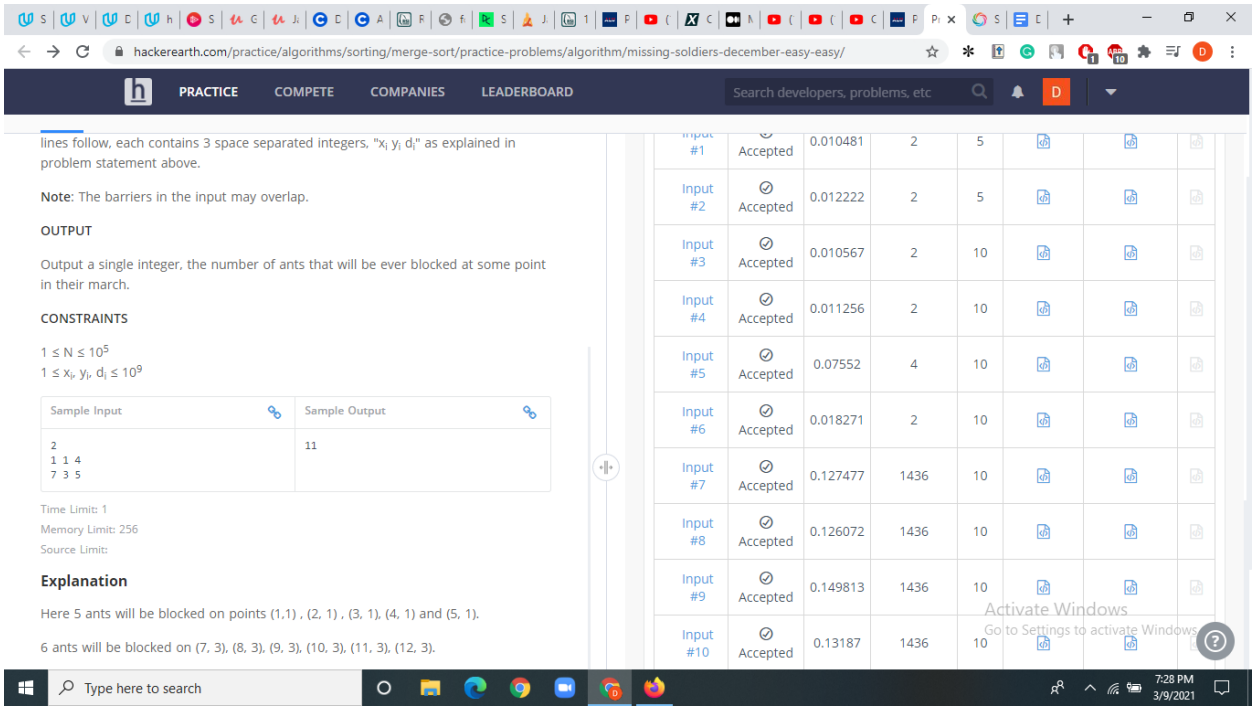
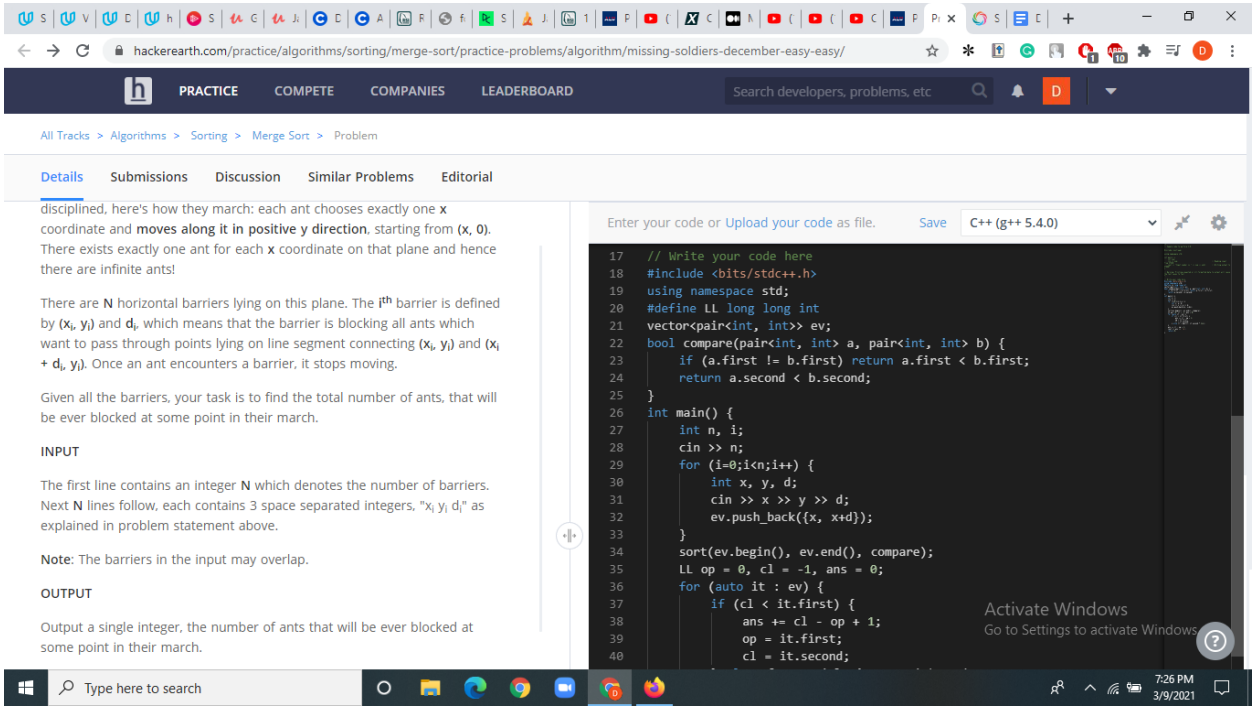
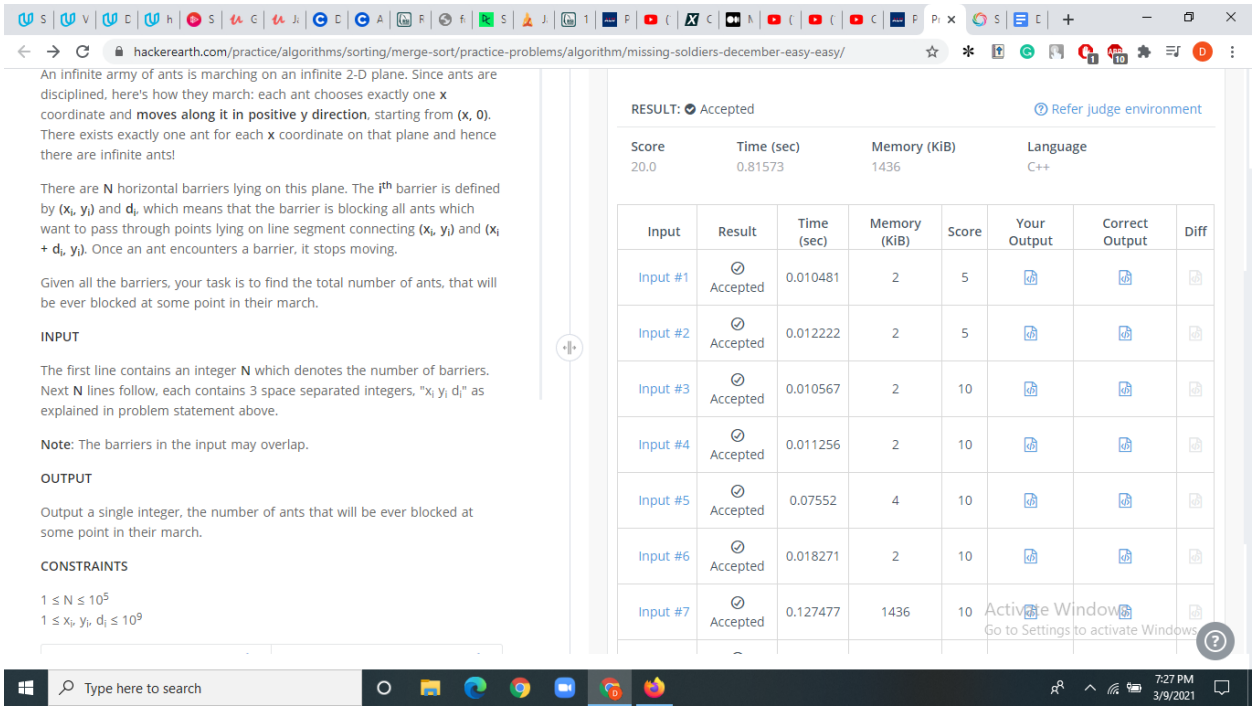

hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/missing-soldiers-december-easy-easy/
PRACTICE
COMPETE
COMPANIES
LEADERBOARD
Search developers, problems, etc
lines follow, each contains 3 space separated integers, "x_i y_i d_i" as explained in problem statement above.
Note: The barriers in the input may overlap.
OUTPUT
Output a single integer, the number of ants that will be ever blocked at some point in their march.
CONSTRAINTS
$1 \le N \le 10^5$
$1 \le x_i, y_i, d_i \le 10^9$
Sample Input
Sample Output
2
1 1 4
7 3 5
11
Time Limit: 1
Memory Limit: 256
Source Limit:
Explanation
Here 5 ants will be blocked on points (1,1) , (2, 1) , (3, 1), (4, 1) and (5, 1).
6 ants will be blocked on (7, 3), (8, 3), (9, 3), (10, 3), (11, 3), (12, 3).
Input #1 Accepted 0.010481 2 5
Input #2 Accepted 0.012222 2 5
Input #3 Accepted 0.010567 2 10
Input #4 Accepted 0.011256 2 10
Input #5 Accepted 0.07552 4 10
Input #6 Accepted 0.018271 2 10
Input #7 Accepted 0.127477 1436 10
Input #8 Accepted 0.126072 1436 10
Input #9 Accepted 0.149813 1436 10
Input #10 Accepted 0.13187 1436 10
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:28 PM
3/9/2021


hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/missing-soldiers-december-easy-easy/
PRACTICE
COMPETE
COMPANIES
LEADERBOARD
Search developers, problems, etc
All Tracks > Algorithms > Sorting > Merge Sort > Problem
Details
Submissions
Discussion
Similar Problems
Editorial
disciplined, here's how they march: each ant chooses exactly one x coordinate and moves along it in positive y direction, starting from (x, 0). There exists exactly one ant for each x coordinate on that plane and hence there are infinite ants!
There are N horizontal barriers lying on this plane. The i^{th} barrier is defined by (x_i, y_i) and d_i, which means that the barrier is blocking all ants which want to pass through points lying on line segment connecting (x_i, y_i) and ($x_i + d_i$, y_i). Once an ant encounters a barrier, it stops moving.
Given all the barriers, your task is to find the total number of ants, that will be ever blocked at some point in their march.
INPUT
The first line contains an integer N which denotes the number of barriers. Next N lines follow, each contains 3 space separated integers, "x_i y_i d_i" as explained in problem statement above.
Note: The barriers in the input may overlap.
OUTPUT
Output a single integer, the number of ants that will be ever blocked at some point in their march.
Enter your code or Upload your code as file.
Save
C++ (g++ 5.4.0)
// Write your code here
#include <bits/stdc++.h>
using namespace std;
#define LL long long int
vector<pair<int, int>> ev;
bool compare(pair<int, int> a, pair<int, int> b) {
if (a.first != b.first) return a.first < b.first;
return a.second < b.second;
}
int main() {
int n, i;
cin >> n;
for (i=0;i<n;i++) {
int x, y, d;
cin >> x >> y >> d;
ev.push_back({x, x+d});
}
sort(ev.begin(), ev.end(), compare);
LL op = 0, cl = -1, ans = 0;
for (auto it : ev) {
if (cl < it.first) {
ans += cl - op + 1;
op = it.first;
cl = it.second;
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:26 PM
3/9/2021



hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/missing-soldiers-december-easy-easy/

An infinite army of ants is marching on an infinite 2-D plane. Since ants are disciplined, here's how they march: each ant chooses exactly one **x** coordinate and **moves along it in positive y direction**, starting from **(x, 0)**. There exists exactly one ant for each **x** coordinate on that plane and hence there are infinite ants!

There are **N** horizontal barriers lying on this plane. The i^{th} barrier is defined by (x_i, y_i) and d_i, which means that the barrier is blocking all ants which want to pass through points lying on line segment connecting (x_i, y_i) and $(x_i + d_i, y_i)$. Once an ant encounters a barrier, it stops moving.

Given all the barriers, your task is to find the total number of ants, that will be ever blocked at some point in their march.

INPUT

The first line contains an integer **N** which denotes the number of barriers. Next **N** lines follow, each contains 3 space separated integers, "x_i y_i d_i" as explained in problem statement above.

Note: The barriers in the input may overlap.

OUTPUT

Output a single integer, the number of ants that will be ever blocked at some point in their march.

CONSTRAINTS

$1 \le N \le 10^5$
$1 \le x_i, y_i, d_i \le 10^9$

RESULT: Accepted — Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
20.0	0.81573	1436	C++

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.010481	2	5			
Input #2	Accepted	0.012222	2	5			
Input #3	Accepted	0.010567	2	10			
Input #4	Accepted	0.011256	2	10			
Input #5	Accepted	0.07552	4	10			
Input #6	Accepted	0.018271	2	10			
Input #7	Accepted	0.127477	1436	10			


```cpp
#include <bits/stdc++.h>
using namespace std;
#define LL long long int
vector<pair<int, int>> ev;
bool compare(pair<int, int> a, pair<int, int> b) {
   if (a.first != b.first) return a.first < b.first;
   return a.second < b.second;
}
int main() {
   int n, i;
   cin >> n;
   for (i=0;i<n;i++) {
       int x, y, d;
       cin >> x >> y >> d;
       ev.push_back({x, x+d});
   }
   sort(ev.begin(), ev.end(), compare);
   LL op = 0, cl = -1, ans = 0;
   for (auto it : ev) {
       if (cl < it.first) {
           ans += cl - op + 1;
           op = it.first;
```

```
            cl = it.second;
        } else cl = max(cl, it.second * 1LL);
    }
    ans += cl - op + 1;
    cout << ans << '\n';
    return 0;
}
```