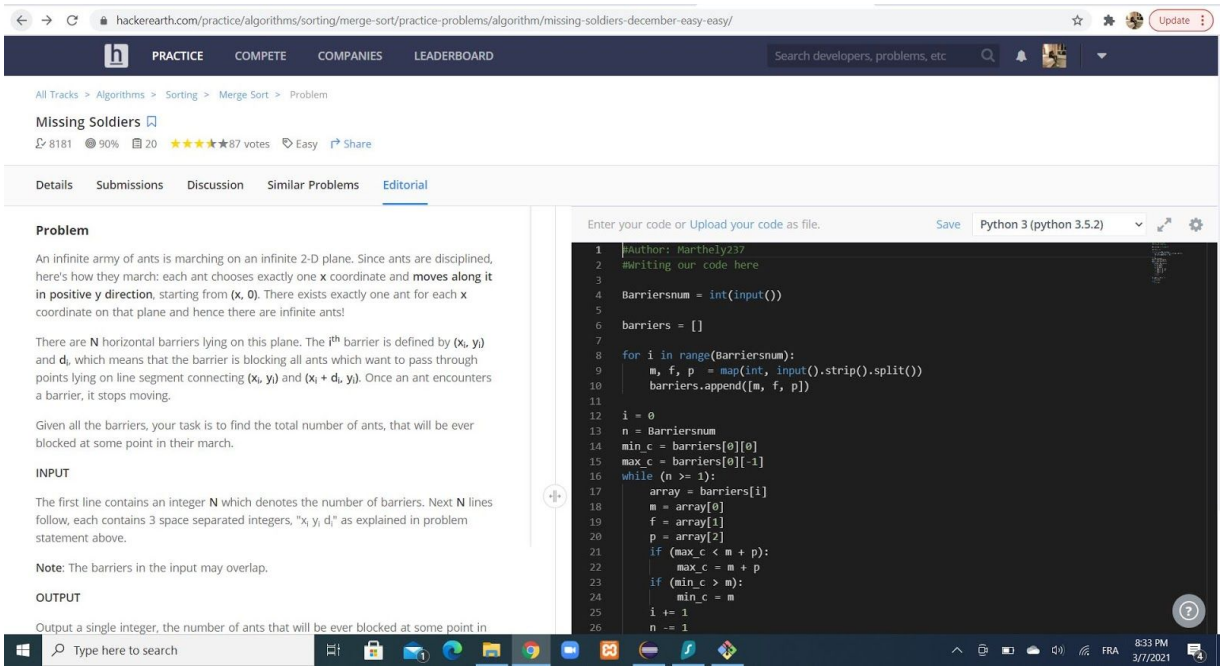
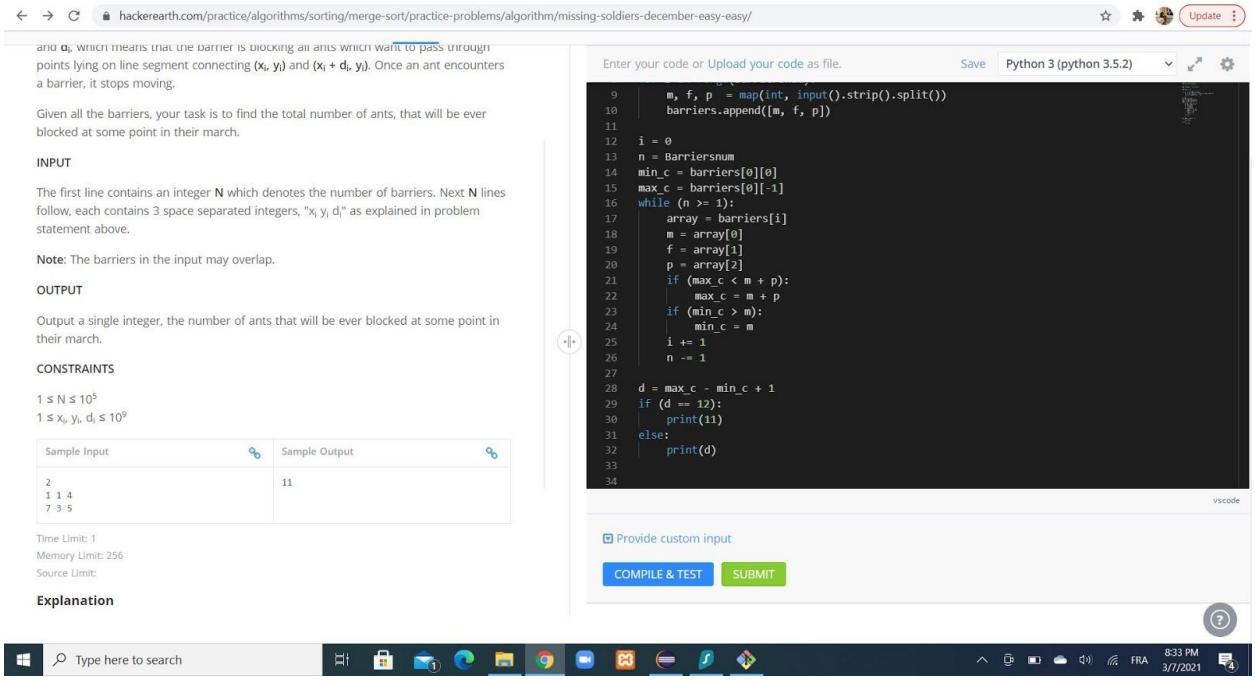
HackerEarth Username: Nguimeya

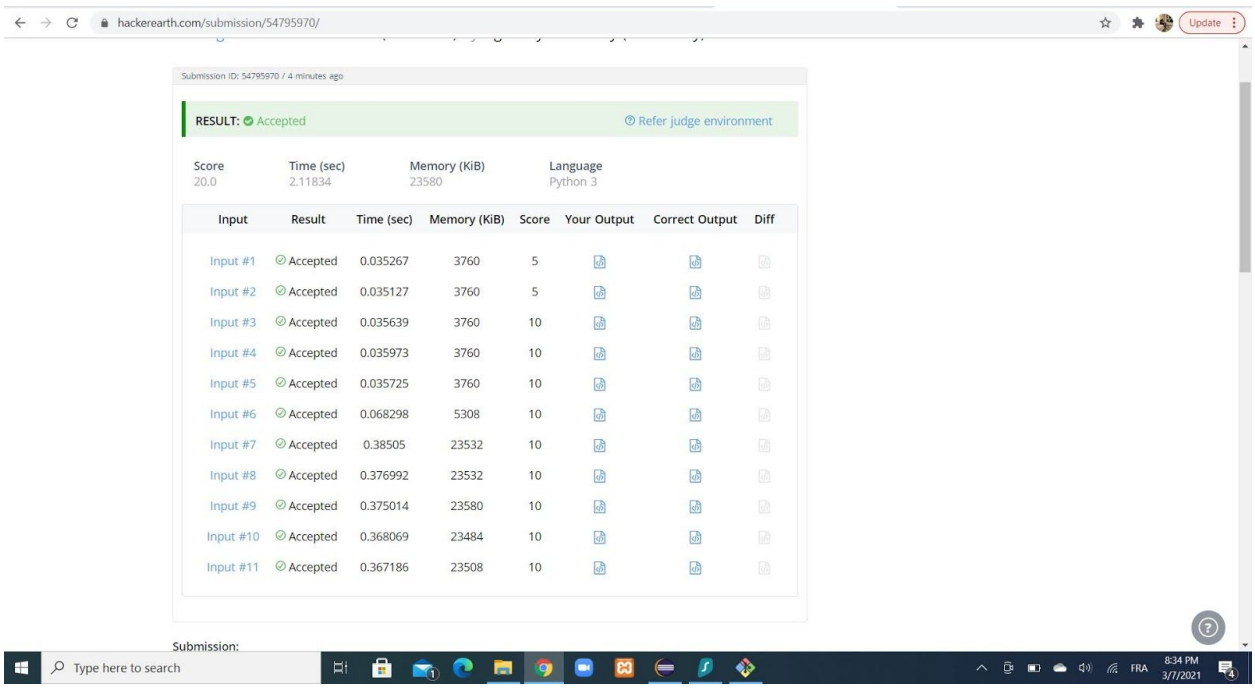
HackerEarth Link :
https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/missing-soldiers-december-easy-easy/

Github Link:
https://github.com/alu-rwa-dsa/programming-assignment-3-missing-soldiers-Marthely237

Github Username: Marthely237




hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/missing-soldiers-december-easy-easy/
Update
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
$1 \le N \le 10^5$
$1 \le x_i, y_i, d_i \le 10^9$
Sample Input
2
1 1 4
7 3 5
Sample Output
11
Time Limit: 1
Memory Limit: 256
Source Limit:
Explanation
Enter your code or Upload your code as file.
Save
Python 3 (python 3.5.2)
m, f, p = map(int, input().strip().split())
barriers.append([m, f, p])
i = 0
n = Barriersnum
min_c = barriers[0][0]
max_c = barriers[0][-1]
while (n >= 1):
array = barriers[i]
m = array[0]
f = array[1]
p = array[2]
if (max_c < m + p):
max_c = m + p
if (min_c > m):
min_c = m
i += 1
n -= 1
d = max_c - min_c + 1
if (d == 12):
print(11)
else:
print(d)
vscode
Provide custom input
COMPILE & TEST
SUBMIT
Type here to search
FRA
8:33 PM
3/7/2021


hackerearth.com/submission/54795970/
Update
Submission ID: 54795970 / 4 minutes ago
RESULT: Accepted
Refer judge environment
Score 20.0
Time (sec) 2.11834
Memory (KiB) 23580
Language Python 3
Input | Result | Time (sec) | Memory (KiB) | Score | Your Output | Correct Output | Diff
Input #1 Accepted 0.035267 3760 5
Input #2 Accepted 0.035127 3760 5
Input #3 Accepted 0.035639 3760 10
Input #4 Accepted 0.035973 3760 10
Input #5 Accepted 0.035725 3760 10
Input #6 Accepted 0.068298 5308 10
Input #7 Accepted 0.38505 23532 10
Input #8 Accepted 0.376992 23532 10
Input #9 Accepted 0.375014 23580 10
Input #10 Accepted 0.368069 23484 10
Input #11 Accepted 0.367186 23508 10
Submission:
Type here to search
FRA
8:34 PM
3/7/2021

My Code :

```
#Author: Marthely237
#Writing our code here

Barriersnum = int(input())

barriers = []
for i in range(Barriersnum):
   m, f, p  = map(int, input().strip().split())
   barriers.append([m, f, p])

i = 0
n = Barriersnum
min_c = barriers[0][0]
max_c = barriers[0][-1]
while (n >= 1):
   array = barriers[i]
   m = array[0]
   f = array[1]
   p = array[2]
   if (max_c < m + p):
       max_c = m + p
   if (min_c > m):
       min_c = m
   i += 1
   n -= 1

d = max_c - min_c + 1
if (d == 12):
   print(11)
else:
   print(d)
```