Falling Trees

locked

Problem Submissions

Time Limit: $C/C{+}{+}\ (2s)$, $Java\ (3s)$
Memory Limit: $256MB$

Lagno is on vacation with her family. They're staying near a forest where they come across a bunch of trees. Her brother notices some trees are standing on a straight line and comes up with the following problem.

There are n trees situated on an infinite line. You intend to cut all the trees. The i^{th} tree is located at an integer coordinate x_i on the line. It is guaranteed that the **trees are located at distinct positions**. The height of the i^{th} tree is a positive integer h_i.

You can cut a tree in one of two ways: left or right, which determines the direction the tree will fall towards once it is cut. A tree at position x_i will fall in the space $[x_i, x_i + h_i]$ inclusive if it is cut on the right side and will fall in the space $[x_i - h_i, x_i]$ inclusive if it is cut on the left side.

To properly cut a tree, the space it falls over in must be empty, that is there cannot be any **other** tree standing there **when it is cut**. Once a tree has fallen, the space it falls in is cleared afterwards and becomes empty. You can assume there is empty space at every location on the infinite line (possibly negative coordinates) that doesn't have a tree.

Let's determine a *bordering space* $[L, R]$, then can cut all the trees in any order and each tree in any way of your choice such that none of the trees fall outside this border. Suppose the length of the *bordering space* is $D = R - L + 1$, what is the minimum D you need to do this?

Since the siblings solved this quickly, their mother proposed a more difficult calculation. She wants the **summation** of $D(x, y)$ over all pairs (x, y) of integers where $1 \leq x \leq y \leq n$. Here $D(x, y)$ is the minimum required length of the *bordering space* if **only** the trees with indices $j\ (x \leq j \leq y)$ existed on the infinite line and the remaining space was empty.

Seeing her brother get lost in the math, she wants your help to figure this out fast, so they can go back to enjoying their holidays.

Input Format

The first line contains an integer t denoting the number of test cases. Then t testcases follow.

The first line of each test case contains an integer n denoting the number of trees in a line.

The second line contains n integers where the i^{th} integer denotes the location x_i of the i^{th} tree.

The third line contains n integers where the i^{th} integer denotes the height h_i of the i^{th} tree.

Constraints

$1 \leq t \leq 3 * 10^5$

$1 \leq x_i, h_i \leq 10^9$

$x_{i-1} < x_i$ for all $i\ (2 \leq i \leq n)$

Sum of n over all testcases does not exceed 10^6

Sum of n^2 over all testcases does not exceed $4 * 10^8$

Output Format

For each testcase print the answer in a separate line.

Sample Input 0

```
2
4
1 7 8 11
1 9 2 3
2
7 9
2 2
```

Sample Output 0

```
78
11
```

Explanation 0

1. First testcase - For $D(1,4)$ bordering space can be in $[-2,11]$.
   - Cut 3^{rd} tree on Right side: Falls in $[8,10]$.
   - Cut 4^{th} tree on Left side: Falls in $[8,11]$.
   - Cut 1^{st} tree on Right side: Falls in $[1,2]$.
   - Cut 2^{nd} tree on Left side: Falls in $[-2,7]$.
2. Second testcase - For $D(1,2)$ bordering space can be in $[7,11]$ by cutting the 2^{nd} & 1^{st} trees in order on the Right. Sum over all pairs is $D(1,1)+D(1,2)+D(2,2)=3+5+3=11$

Submissions: 0
Max Score: 1

Rate This Challenge:

More

C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File ☐ Test against custom input Run Code Submit Code