Topic Stream Mashup: DP Optimizations (segment tree, convex hull trick)

A. Introduction: Segment Tree Optimization

2 seconds, 256 megabytes

An introduction to optimizing with segment tree. Problem source: me

You have a line with n locations, where each location i has a value a_i corresponding to it. Define $s(l, r)$ to be $\sum_{i=l}^{r} a_i$: the sum of values for locations between l and r. You can jump from any position i to any position j where $i < j$ for a cost of $s(i, j)$.

However, some locations are quite selective in where you can come from. In particular, each location j has values l_j and r_j corresponding to it, and you can only jump to location j from locations i where $l_j \le i \le r_j$. For each i from 1 to n, find the minimum cost to get from location 1 to location i.

Input

The first line of input contains a single integer n $(1 \le n \le 2 \cdot 10^5)$, the number of locations.

The next line contains n integers, representing the values a_i $(-10^6 \le a_i \le 10^6)$.

The i-th line of the following $n - 1$ lines contain the values l_{i+1} and r_{i+1} $(1 \le l_{i+1} \le r_{i+1} < i + 1)$ [so, $2 \le i \le n$, since l_1 and r_1 would be meaningless].

Output

Output n integers, the minimum cost to get from location 1 to location i for each i from 1 to n.

input

```
4
1 2 3 4
1 1
1 2
2 3
```

output

```
0 3 6 12
```

input

```
6
5 7 -12 9 -3 -7
1 1
2 2
1 3
2 4
4 5
```

output

```
0 12 7 4 1 -9
```

input

```
10
51 -27 38 44 -99 71 -80 91 38 23
1 1
1 2
2 3
1 2
3 5
2 3
6 7
5 6
6 9
```

output

```
0 24 35 79 -20 -48 -29 -18 1 43
```

For the first sample:

The answer for $i = 1$ is 0, because you start at 1.

The answer for $i = 2$ is 3, where you jump from 1 to 2 (cost 3).

The answer for $i = 3$ is 6, where you jump from 1 to 3 (cost 6).

The answer for $i = 4$ is 12, where you jump from 1 to 2 (cost 3), then 2 to 4 (cost 9). The alternative is going $1 \to 3 \to 4$, which gives you a cost of 13, which is worse.

B. Introduction: Convex Hull Trick Optimization

2 seconds, 256 megabytes

An introduction to optimizing with convex hull trick. Problem source: me

You have a line with n locations, where each location i has values a_i and b_i corresponding to it. Define $s(l, r)$ to be $\sum_{i=l}^{r} a_i$: the sum of values for locations between l and r. You can jump from any position i to any position j where $i < j$ for a cost of $s(i, j) \cdot b_j$. For each i from 1 to n, find the minimum cost to get from location 1 to location i.

In this problem, the values l_j and r_j don't exist (don't worry, I won't make you write a segment tree over convex hulls... yet). There's no restriction on jumping, only that you can't jump backwards (so, to jump from i to j, $i < j$ is required).

Input

The first line of input contains a single integer n $(1 \le n \le 2 \cdot 10^5)$, the number of locations.

The next line contains n integers, representing the values a_i $(-10^6 \le a_i \le 10^6)$.

The next line contains n integers, representing the values b_i $(-10^6 \le a_i \le 10^6)$.

Output

Output n integers, the minimum cost to get from location 1 to location i for each i from 1 to n.

input

```
4
1 2 3 4
1 2 3 4
```

output

```
0 6 18 40
```

input

```
6
5 7 -12 9 -3 -7
3 -11 2 6 8 -1
```

output

```
0 -132 -142 -160 -190 -180
```

input

```
10
51 -27 38 44 -99 71 -80 91 38 23
77 15 -36 -40 99 -65 28 69 70 71
```

output

```
0 360 -2232 -5512 -10957 -9137 -13981 -13222 -10551 -8869
```

For the first sample:

The answer for $i = 1$ is 0, because you start at 1.

The answer for $i = 2$ is 6, where you jump from 1 to 2 (cost 6).

The answer for $i = 3$ is 18, where you jump from 1 to 3 (cost 18).

The answer for $i = 4$ is 40, where you jump from 1 to 4 (cost 40).

C. Subsequences

3.0 s, 256 MB

Problem source: https://codeforces.com/problemset/problem/597/C

For the given sequence with n different elements find the number of increasing subsequences with $k + 1$ elements. It is guaranteed that the answer is not greater than $8 \cdot 10^{18}$.

Input

First line contain two integer values n and k $(1 \le n \le 10^5, 0 \le k \le 10)$ — the length of sequence and the number of elements in increasing subsequences.

Next n lines contains one integer a_i ($1 \le a_i \le n$) each — elements of sequence. All values a_i are different.

Output

Print one integer — the answer to the problem.

input
5 2 1 2 3 5 4

output
7

D. Hanoi Factory

2.0 s, 256 MB

Problem source: https://codeforces.com/problemset/problem/777/E

Of course you have heard the famous task about Hanoi Towers, but did you know that there is a special factory producing the rings for this wonderful game? Once upon a time, the ruler of the ancient Egypt ordered the workers of Hanoi Factory to create as high tower as possible. They were not ready to serve such a strange order so they had to create this new tower using already produced rings.

There are n rings in factory's stock. The i-th ring has inner radius a_i, outer radius b_i and height h_i. The goal is to select some subset of rings and arrange them such that the following conditions are satisfied:

- Outer radiuses form a non-increasing sequence, i.e. one can put the j-th ring on the i-th ring only if $b_j \le b_i$.
- Rings should not fall one into the the other. That means one can place ring j on the ring i only if $b_j > a_i$.
- The total height of all rings used should be maximum possible.

Input

The first line of the input contains a single integer n ($1 \le n \leqslant 100\,000$) — the number of rings in factory's stock.

The i-th of the next n lines contains three integers a_i, b_i and h_i ($1 \le a_i, b_i, h_i \le 10^9$, $b_i > a_i$) — inner radius, outer radius and the height of the i-th ring respectively.

Output

Print one integer — the maximum height of the tower that can be obtained.

input
3 1 5 1 2 6 2 3 7 3

output
6

input
4 1 2 1 1 3 3 4 6 2 5 7 1

output
4

In the first sample, the optimal solution is to take all the rings and put them on each other in order 3, 2, 1.

In the second sample, one can put the ring 3 on the ring 4 and get the tower of height 3, or put the ring 1 on the ring 2 and get the tower of height 4.

E. Kalila and Dimna in the Logging Industry

2 s., 256 MB

Problem source: https://codeforces.com/problemset/problem/319/C

Kalila and Dimna are two jackals living in a huge jungle. One day they decided to join a logging factory in order to make money.

The manager of logging factory wants them to go to the jungle and cut n trees with heights $a_1, a_2, \ldots, a_n$. They bought a chain saw from a shop. Each time they use the chain saw on the tree number i, they can decrease the height of this tree by one unit. Each time that Kalila and Dimna use the chain saw, they need to recharge it. Cost of charging depends on the id of the trees which have been cut completely (a tree is cut completely if its height equal to 0). If the maximum id of a tree which has been cut completely is i (the tree that have height a_i in the beginning), then the cost of charging the chain saw would be b_i. If no tree is cut completely, Kalila and Dimna cannot charge the chain saw. The chainsaw is charged in the beginning. We know that for each $i < j$, $a_i < a_j$ and $b_i > b_j$ and also $b_n = 0$ and $a_1 = 1$. Kalila and Dimna want to cut all the trees completely, with minimum cost.

They want you to help them! Will you?

Input

The first line of input contains an integer n ($1 \le n \le 10^5$). The second line of input contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$). The third line of input contains n integers $b_1, b_2, \ldots, b_n$ ($0 \le b_i \le 10^9$).

It's guaranteed that $a_1 = 1$, $b_n = 0$, $a_1 < a_2 < \cdots < a_n$ and $b_1 > b_2 > \cdots > b_n$.

Output

The only line of output must contain the minimum cost of cutting all the trees completely.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

input
5 1 2 3 4 5 5 4 3 2 0

output
25

input
6 1 2 3 10 20 30 6 5 4 3 2 0

output
138

F. Pencils and Boxes

2 s., 256 MB

Problem source: https://codeforces.com/problemset/problem/985/E

Mishka received a gift of multicolored pencils for his birthday! Unfortunately he lives in a monochrome world, where everything is of the same color and only saturation differs. This pack can be represented as a sequence $a_1, a_2, \ldots, a_n$ of n integer numbers — saturation of the color of each pencil. Now Mishka wants to put all the mess in the pack in order. He has an infinite number of empty boxes to do this. He would like to fill some boxes in such a way that:

- Each pencil belongs to **exactly** one box;
- Each non-empty box has at least k pencils in it;
- If pencils i and j belong to the same box, then $|a_i - a_j| \le d$, where $|x|$ means absolute value of x. Note that the opposite is optional, there can be pencils i and j such that $|a_i - a_j| \le d$ and they belong to different boxes.

Help Mishka to determine if it's possible to distribute all the pencils into boxes. Print "`YES`" if there exists such a distribution. Otherwise print "`NO`".

Input

The first line contains three integer numbers n, k and d ($1 \le k \le n \le 5 \cdot 10^5$, $0 \le d \le 10^9$) — the number of pencils, minimal size of any non-empty box and maximal difference in saturation between any pair of pencils in the same box, respectively.

The second line contains n integer numbers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — saturation of color of each pencil.

Output
Print "YES" if it's possible to distribute all the pencils into boxes and satisfy all the conditions. Otherwise print "NO".

input
6 3 10 7 2 7 7 4 2
output
YES

input
6 2 3 4 5 3 13 4 10
output
YES

input
3 2 5 10 16 22
output
NO

In the first example it is possible to distribute pencils into 2 boxes with 3 pencils in each with any distribution. And you also can put all the pencils into the same box, difference of any pair in it won't exceed 10.

In the second example you can split pencils of saturations $[4, 5, 3, 4]$ into 2 boxes of size 2 and put the remaining ones into another box.

G. Trains and Statistic

2 s., 256 MB

Problem source: https://codeforces.com/problemset/problem/675/E

Vasya commutes by train every day. There are n train stations in the city, and at the i-th station it's possible to buy only tickets to stations from $i + 1$ to a_i inclusive. No tickets are sold at the last station.

Let $\rho_{i,j}$ be the minimum number of tickets one needs to buy in order to get from stations i to station j. As Vasya is fond of different useless statistic he asks you to compute the sum of all values $\rho_{i,j}$ among all pairs $1 \le i < j \le n$.

Input
The first line of the input contains a single integer n ($2 \le n \le 100\,000$) — the number of stations.

The second line contains $n - 1$ integer a_i ($i + 1 \le a_i \le n$), the i-th of them means that at the i-th station one may buy tickets to each station from $i + 1$ to a_i inclusive.

Output
Print the sum of $\rho_{i,j}$ among all pairs of $1 \le i < j \le n$.

input
4 4 4 4
output
6

input
5 2 3 5 5
output
17

In the first sample it's possible to get from any station to any other (with greater index) using only one ticket. The total number of pairs is 6, so the answer is also 6.

Consider the second sample:

- $\rho_{1,2} = 1$
- $\rho_{1,3} = 2$
- $\rho_{1,4} = 3$
- $\rho_{1,5} = 3$
- $\rho_{2,3} = 1$
- $\rho_{2,4} = 2$
- $\rho_{2,5} = 2$
- $\rho_{3,4} = 1$
- $\rho_{3,5} = 1$
- $\rho_{4,5} = 1$

Thus the answer equals $1 + 2 + 3 + 3 + 1 + 2 + 2 + 1 + 1 + 1 = 17$.

H. The Fair Nut and Rectangles

5.0 s, 256 MB

Problem source: https://codeforces.com/problemset/problem/1083/E

The Fair Nut got stacked in planar world. He should solve this task to get out.

You are given n rectangles with vertexes in $(0, 0)$, $(x_i, 0)$, (x_i, y_i), $(0, y_i)$. For each rectangle, you are also given a number a_i. Choose some of them that the area of union minus sum of a_i of the chosen ones is maximum.

It is guaranteed that there are no nested rectangles.

Nut has no idea how to find the answer, so he asked for your help.

Input
The first line contains one integer n ($1 \le n \le 10^6$) — the number of rectangles.

Each of the next n lines contains three integers x_i, y_i and a_i ($1 \le x_i, y_i \le 10^9$, $0 \le a_i \le x_i \cdot y_i$).

It is guaranteed that there are no nested rectangles.

Output
In a single line print the answer to the problem — the maximum value which you can achieve.

input
3 4 4 8 1 5 0 5 2 10
output
9

input
4 6 2 4 1 6 2 2 4 3 5 3 8
output
10

In the first example, the right answer can be achieved by choosing the first and the second rectangles.

In the second example, the right answer can also be achieved by choosing the first and the second rectangles.

I. Escape Through Leaf

3 s., 256 MB

Problem source: https://codeforces.com/problemset/problem/932/F

You are given a tree with n nodes (numbered from 1 to n) rooted at node 1. Also, each node has two values associated with it. The values for i-th node are a_i and b_i.

You can jump from a node to any node in its subtree. The cost of one jump from node x to node y is the product of a_x and b_y. The total cost of a path formed by one or more jumps is sum of costs of individual jumps. For every node, calculate the minimum total cost to reach any leaf from that node. Pay attention, that root can never be leaf, even if it has degree 1.

Note that you cannot jump from a node to itself.

Input

The first line of input contains an integer n ($2 \le n \le 10^5$) — the number of nodes in the tree.

The second line contains n space-separated integers $a_1,\ a_2,\ \cdots,\ a_n (-10^5 \le a_i \le 10^5)$.

The third line contains n space-separated integers $b_1,\ b_2,\ \cdots,\ b_n (-10^5 \le b_i \le 10^5)$.

Next $n - 1$ lines contains two space-separated integers u_i and v_i ($1 \le u_i,\ v_i \le n$) describing edge between nodes u_i and v_i in the tree.

Output

Output n space-separated integers, i-th of which denotes the minimum cost of a path from node i to reach any leaf.

input

```
3
2 10 -1
7 -7 5
2 3
2 1
```

output

```
10 50 0
```

input

```
4
5 -10 5 7
-8 -80 -3 -10
2 1
2 4
1 3
```

output

```
-300 100 0 0
```

In the first example, node 3 is already a leaf, so the cost is 0. For node 2, jump to node 3 with cost $a_2 \times b_3 = 50$. For node 1, jump directly to node 3 with cost $a_1 \times b_3 = 10$.

In the second example, node 3 and node 4 are leaves, so the cost is 0. For node 2, jump to node 4 with cost $a_2 \times b_4 = 100$. For node 1, jump to node 2 with cost $a_1 \times b_2 = -400$ followed by a jump from 2 to 4 with cost $a_2 \times b_4 = 100$.

J. Cookie Clicker

2 s., 256 MB

Problem source: https://codeforces.com/problemset/problem/377/E

Kostya is playing the computer game Cookie Clicker. The goal of this game is to gather cookies. You can get cookies using different *buildings*: you can just click a special field on the screen and get the cookies for the clicks, you can buy a cookie factory, an alchemy lab, a time machine and it all will bring lots and lots of cookies.

At the beginning of the game (time 0), Kostya has 0 cookies and no buildings. He has n available buildings to choose from: the i-th building is worth c_i cookies and when it's built it brings v_i cookies at the end of each second. Also, to make the game more interesting to play, Kostya decided to add a limit: at each moment of time, he can use only one building. Of course, he can change the active building each second at his discretion.

It's important that Kostya is playing a version of the game where he can buy new buildings and change active building only at time moments that are multiples of one second. Kostya can buy new building and use it at the same time. If Kostya starts to use a building at the time moment t, he can get the first profit from it only at the time moment $t + 1$.

Kostya wants to earn at least s cookies as quickly as possible. Determine the number of seconds he needs to do that.

Input

The first line contains two integers n and s ($1 \le n \le 2 \cdot 10^5$, $1 \le s \le 10^{16}$) — the number of buildings in the game and the number of cookies Kostya wants to earn.

Each of the next n lines contains two integers v_i and c_i ($1 \le v_i \le 10^8$, $0 \le c_i \le 10^8$) — the number of cookies the i-th building brings per second and the building's price.

Output

Output the only integer — the minimum number of seconds Kostya needs to earn at least s cookies. It is guaranteed that he can do it.

input

```
3 9
1 0
2 3
5 4
```

output

```
6
```

input

```
3 6
1 0
2 2
5 4
```

output

```
5
```

input

```
3 13
1 0
2 2
6 5
```

output

```
7
```

input

```
1 10000000000000000
1 0
```

output

```
10000000000000000
```

K. Function

2.0 s, 256 MB

Okay, so maybe this one isn't technically DP. But it's the culmination of everything relevant to this mashup, and I think it's a good "final boss".
Problem source: https://codeforces.com/problemset/problem/455/E

Serega and Fedor play with functions. One day they came across a very interesting function. It looks like that:

- $f(1, j) = a[j], 1 \le j \le n$.
- $f(i, j) = \min(f(i - 1, j), f(i - 1, j - 1)) + a[j], 2 \le i \le n, i \le j \le n$.

Here a is an integer array of length n.

Serega and Fedya want to know what values this function takes at some points. But they don't want to calculate the values manually. So they ask you to help them.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the length of array a. The next line contains n integers: $a[1], a[2], \ldots, a[n]$ ($0 \le a[i] \le 10^4$).

The next line contains integer m ($1 \le m \le 10^5$) — the number of queries. Each of the next m lines contains two integers: x_i, y_i ($1 \le x_i \le y_i \le n$). Each line means that Fedor and Serega want to know the value of $f(x_i, y_i)$.

Output

Print m lines — the answers to the guys' queries.

input

```
6
2 2 3 4 3 4
4
4 5
3 4
3 4
2 3
```

output

```
12
9
9
5
```

input
7 1 3 2 3 4 0 2 4 4 5 2 3 1 4 4 6

output
11 4 3 0