Task Vacation

⏳1.2 sec. 💾 512 MB

Anton şi prietenii lui planifică o vacanţă împreună. Ei au ales deja locaţia; cu toate acestea, datele sunt mai greu de convenit.

All N friends have submitted in advance the days they plan to take off work. Friend i originally scheduled their time off from day L_i to day R_i, inclusive. To maximize the time they can spend together, each friend may adjust their time off by shifting it earlier or later. Specifically, the i-th friend can choose an integer d_i and move their time off to the interval $[L_i + d_i, R_i + d_i]$. A positive d_i means taking time off later than originally planned, a negative d_i means earlier, and $d_i = 0$ means keeping the original schedule.

The friends recognize that their bosses will not like the disruption caused by their changes. Therefore, they will only move their days off in such a way that the sum $|d_0| + |d_1| + \cdots + |d_{N-1}|$ does not exceed some integer K.

Help the friends figure out the maximum number of days **all of them** can be together if they change their schedules optimally.

Implementation details

You should implement the function `plan_vacation`:

```
int plan_vacation(int N, std::vector<int> L, std::vector<int> R, long
    long K)
```

- N: the number of friends
- L: a vector of N positive integers, each of which denotes the first day off for that friend;
- R: a vector of N positive integers, each of which denotes the last day off for that friend;
- K: the maximum allowed value of $|d_0| + |d_1| + \cdots + |d_{N-1}|$.

This function will be called once for each test. It has to return the maximum number of days all friends can be together or 0 if that isn't possible at all.

Constraints

- $1 \le N \le 500\,000$
- $1 \le L_i \le R_i \le 10^9$
- $0 \le K \le 10^{18}$

Subtasks

Subtask	Points	Required subtasks	Additional constraints
0	0	–	Example.
1	7	–	$K = 0$
2	11	1	$K \le 1$
3	6	–	$K = 10^{18}$
4	13	–	$N \le 10^4$, $L_i \le 10$, $R_i \le 10$
5	18	–	$N \le 10^3$
6	29	$4, 5$	$N \le 10^5$
7	16	$1 - 6$	–

Example

Consider the following call:

```
plan_vacation(3, {1, 5, 2}, {3, 9, 5}, 3)
```

The friends have requested the following intervals of days off: $[1, 3]$, $[5, 9]$, $[2, 5]$. Therefore, we can move friend 0 time off later by 2 days and friend 1 time off earlier by 1 day to get $[3, 5]$, $[4, 8]$, $[2, 5]$. Then, all friends would be available on day 4 and day 5, which results in 2 days in common. It can be proven that we can't do better with $K = 3$. The function should return 2.

Sample grader

The input format is the following:

- line 1: two integers – the values of N and K.
- lines 2 to $N + 1$: two integers – L_i and R_i.

The output format is the following:

- line 1: one integer – the return value of the call.