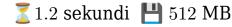


#### Task Odmor



Nakon 1. dana natjecanja u kojemu Borna nije pronašao put do 2 boda, Borna se nalazi u depresiji. Za oporavak od depresije, Borna je odlučio otići na odmor s imaginarnim prijateljima s kojima je dijelio bod. Izabrali su lokaciju MIOC, ali ne mogu se dogovoriti oko datuma.

Svih N imaginarnih prijatelja je odredilo kojim danima žele posjetiti MIOC. Imaginarni prijatelj i želi posjetiti najbolju školu svakog dana između  $L_i$  i  $R_i$ , uključivo. Da maksimiziraju vrijeme koje će provesti zajedno, svaki imaginarni prijatelj može pomaknutu svoje vrijeme za nekoliko dana. Toćnije, i-ti imaginarni prijatelj može izabrati broj  $d_i$  i pomaknuti svoj interval na  $[L_i+d_i,R_i+d_i]$ . Pozitivni  $d_i$  znači otići kasnije od zamišljenog, negativni  $d_i$  ranije, a  $d_i=0$  znači nepromijenjen interval.

Imaginarni prijatelji znaju da ne smiju napraviti prevelike promjene jer će tada Borna poludjeti od viška informacija. Zato mogu pomaknuti svoje željene dane tako da suma  $|d_0| + |d_1| + \cdots + |d_{N-1}|$  ne bude veća od broja K.

Pomognite imaginarnim prijateljima i borni odrediti maksimalan broj dana(nadamo se više nego bodova) takvih da **svi** mogu biti zajedno ako promjene svoje želje optimalno.

# **1** Implementacijski detalji

Trebaš implementirati funkciju plan\_vacation:

int plan\_vacation(int N, std::vector<int> L, std::vector<int> R,
long long K)

- N: broj imaginarnih prijatelja
- L: vektor N pozitivnih brojeva, svaki označava prvi dan u želji imaginarnog prijatelja
- ullet R: vektor N pozitivnih brojeva, svaki označava zadnji dan u želji imaginarnog prijatelja
- K: maksimalna dozvoljena vrijednost od  $|d_0| + |d_1| + \cdots + |d_{N-1}|$ .

Ova funkcija biti će pozvana jednom po primjeru. Treba vratiti maksimalan broj dana takvih da svi imaginarni prijatelji mogu biti zajedno ili 0 ako to nije moguće



## Ograničenja

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

#### **Podzadaci**

| Podzadatak | Bodovi | Potrebni podzadaci | Dodatna ograničenja                           |
|------------|--------|--------------------|---|
| 0          | 0      | _                  | Primjer.                                      |
| 1          | 7      | _                  | K = 0   |
| 2          | 11     | 1                  | $K \leq 1$                                    |
| 3          | 6      | _                  | $K = 10^{18}$                                 |
| 4          | 13     | 0                  | $N \leq 10^4$ , $L_i \leq 10$ , $R_i \leq 10$ |
| 5          | 18     | 0                  | $N \le 10^3$                                  |
| 6          | 29     | 0, 4, 5            | $N \le 10^5$                                  |
| 7          | 16     | 0 - 6              | _   |

# Primjer

Promatrajmo ovaj poziv:

Želje prijatelja su redom intervali: [1, 3], [5, 9], [2, 5]. Možemo pomaknuti vrijeme prijatelja 0 za 2 dana kasnije, prijatelja 1 za 1 dan ranije da dobijemo intervale [3, 5], [4, 8], [2, 5]. Tada su svi prijatelji u MIOCU mna dan 4 i dan 5, što rezultira da su na 2 dana svi u MIOCU. Ne postoji rješenje veće od navedenoga pa funkcija vraća 2.

### Ocjenjivač primjera

Input je u formatu:

- linija 1: 2 broja vrijednosti od *N* i *K*.
- linije 2 do N+1: brojevi  $L_i$  and  $R_i$ .

Output je u formatu:

• linija 1: 1 broj - povratna vrijednost funkcije.