



Среци

N планини, номерирани от 0 до $N - 1$ от ляво на дясно, лежат на хоризонтална линия. Височината на i -тата планина е H_i ($0 \leq i \leq N - 1$). На върха на всяка от планините живее по 1 човек.

Трябва да организирате Q срещи, номерирани от 0 до $Q - 1$. В срещата с номер j ($0 \leq j \leq Q - 1$) ще участват хората, които живеят на планини с номера от L_j до R_j включително ($0 \leq L_j \leq R_j \leq N - 1$). За провеждането на тази среща трябва да се избере планина x , като ($L_j \leq x \leq R_j$). Цената на срещата се пресмята по следния начин:

- Цената на срещата е сума на цените, които трябва да платят участниците.
- Цената, която трябва да плати участникът, живеещ на планината с номер y ($L_j \leq y \leq R_j$) е равна на височината на най-високата планина, намираща се в интервала от x до y , включително.
- В частност, цената за участника живеещ на планината x е H_x - височината на планината x .

За всяка среща трябва да изберете място на провеждане така че цената да е минимална.

Забележете, че след завършването на поредната среща, всички участници се завръщат на своята планина, така че цената на всяка среща не се влияе от мястото на провеждане на предната.

Детайли за реализацията

Напишете следната функция:

```
int64[] minimum_costs(int[] H, int[] L, int[] R)
```

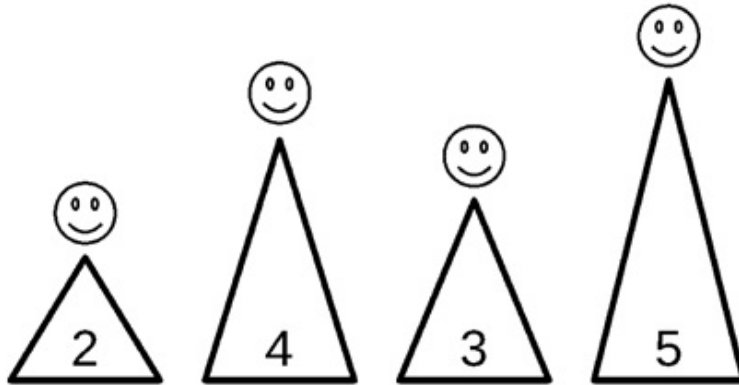
- H : масив с дължина N , съдържащ височините на планините.
- L и R : масиви с дължина Q , съдържащи началата и краищата на интервалите от участници за съответните срещи.
- Функцията трябва да връща намерените стойности в масив (да го наречем условно C) с дължина Q . Стойността C_j ($0 \leq j \leq Q - 1$) трябва да е минималната възможна цена за провеждане на j -тата среща.
- Как да получите дължините N и Q на масивите H и C е описано в таблицата

на страница "Бележки".

Пример

Нека $N = 4$, $H = [2, 4, 3, 5]$, $Q = 2$, $L = [0, 1]$, и $R = [2, 3]$.

Грейдърът извиква `minimum_costs([2, 4, 3, 5], [0, 1], [2, 3])`.



Срещата $j = 0$ е в интервала $L_j = 0$ and $R_j = 2$, така че ще бъде посетена от живеещите на планините 0, 1, и 2. Ако изберете планината 0 като място на срещата, цената на срещата се пресмята както следва:

- Цената на участника от планината 0 е $\max\{H_0\} = 2$.
- Цената на участника от планината 1 е $\max\{H_0, H_1\} = 4$.
- Цената на участника от планината 2 е $\max\{H_0, H_1, H_2\} = 4$.
- Следователно, цената на среща 0 е $2 + 4 + 4 = 10$.

Не е възможно да се проведе срещата 0 на друго място с по-ниска цена, затова цената на тази среща е 10.

Срещата $j = 1$ е за интервала от $L_j = 1$ до $R_j = 3$ и в нея ще участват живеещите на планините 1, 2 и 3. Ако изберем за място на срещата планината 2, цената на тази среща се пресмята както следва::

- Цената на участника от планината 1 е $\max\{H_1, H_2\} = 4$.
- Цената на участника от планината 2 е $\max\{H_2\} = 3$.
- Цената на участника от планината 3 е $\max\{H_2, H_3\} = 5$.
- Следователно, цената на среща 1 е $4 + 3 + 5 = 12$.

Не е възможно се проведе среща 1 на друго място при по-ниска цена, затова цената на тази среща е 12.

Файловете `sample-01-in.txt` и `sample-01-out.txt` в прикачения архив съответстват на този пример. В архива ще намерите и други примерни входове/изходи.

Ограничения

- $1 \leq N \leq 750\,000$
- $1 \leq Q \leq 750\,000$
- $1 \leq H_i \leq 1\,000\,000\,000$ ($0 \leq i \leq N - 1$)
- $0 \leq L_j \leq R_j \leq N - 1$ ($0 \leq j \leq Q - 1$)
- $(L_j, R_j) \neq (L_k, R_k)$ ($0 \leq j < k \leq Q - 1$)

Подзадачи

1. (4 точки) $N \leq 3\,000$, $Q \leq 10$
2. (15 точки) $N \leq 5\,000$, $Q \leq 5\,000$
3. (17 точки) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 2$ ($0 \leq i \leq N - 1$)
4. (24 точки) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 20$ ($0 \leq i \leq N - 1$)
5. (40 points) Няма допълнителни ограничения.

Примерен грейдър

Примерният грейдър чете вход в следния формат:

- ред 1: N Q
- ред 2: H_0 H_1 \dots H_{N-1}
- ред 3 + j ($0 \leq j \leq Q - 1$): L_j R_j

Примерният грейдър извежда стойностите, върнати от функцията `minimum_costs`, в следния формат:

- ред 1 + j ($0 \leq j \leq Q - 1$): C_j