

Ніл

Ви хочете перевезти N артефактів через Ніл. Артефакти пронумеровані від 0 до $N - 1$. Вага артефакту i ($0 \leq i < N$) становить $W[i]$.

Для транспортування артефактів ви використовуєте спеціальні човни. Кожен човен може перевозити **не більше двох** артефактів.

- Якщо ви вирішили помістити один артефакт у човен, вага артефакту може бути довільною.
- Якщо ви хочете помістити два артефакти в один човен, ви повинні переконатися, що човен рівномірно збалансований. Зокрема, можна відправити артефакти p та q ($0 \leq p < q < N$) в одному човні тільки тоді, коли абсолютна різниця між їхніми вагами не перевищує D , тобто $|W[p] - W[q]| \leq D$.

Щоб транспортувати артефакт, ви повинні заплатити суму, яка залежить від кількості артефактів, які перевозяться в одному човні. Вартість транспортування артефакту i ($0 \leq i < N$) становить:

- $A[i]$, якщо ви помістите один артефакт у човен, або
- $B[i]$, якщо ви помістите його в човен разом з іншим артефактом.

Зверніть увагу, що в останньому випадку вам доведеться заплатити за обидва артефакти в човні. Зокрема, якщо ви вирішите відправити артефакти p та q ($0 \leq p < q < N$) в одному човні, вам потрібно заплатити $B[p] + B[q]$.

Відправити артефакт на човні окремо завжди дорожче ніж надіслати його з іншим артефактом в одному човні, тому $B[i] < A[i]$ для усіх i таких, що $0 \leq i < N$.

На жаль, річка дуже непередбачувана, і значення D часто змінюється. Ваше завдання — відповісти на Q запитів з номерами від 0 до $Q - 1$. Запити описуються масивом E довжини Q . Відповідь на запит j ($0 \leq j < Q$) – це мінімальна загальна вартість транспортування всіх N артефактів, коли значення D дорівнює $E[j]$.

Деталі реалізації

Ви повинні реалізувати наступну функцію.

```
std::vector<long long> calculate_costs(
    std::vector<int> W, std::vector<int> A,
    std::vector<int> B, std::vector<int> E)
```

- W, A, B : масиви цілих чисел довжини N , що описують вагу артефактів і вартість їх транспортування.
- E : масив цілих чисел довжини Q , що описує значення D для кожного запиту.
- Ця функція має повернути масив R з Q цілих чисел, що містить мінімальну загальну вартість транспортування артефактів, де $R[j]$ задає вартість, коли значення D дорівнює $E[j]$ (для кожного j такого, що $0 \leq j < Q$).
- Ця функція викликається рівно один раз для кожного тесту.

Обмеження

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ для кожного i такого, що $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$ для кожного i такого, що $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$ для кожного j такого, що $0 \leq j < Q$

Підзадачі

Підзадача	Балів	Додаткові обмеження
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ для кожного i такого, що $0 \leq i < N$
2	13	$Q \leq 5; W[i] = i + 1$ для кожного i такого, що $0 \leq i < N$
3	17	$Q \leq 5; A[i] = 2 \text{ і } B[i] = 1$ для кожного i такого, що $0 \leq i < N$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2 \text{ і } B[i] = 1$ для кожного i такого, що $0 \leq i < N$
7	18	Без додаткових обмежень

Приклад

Розглянемо наступний виклик.

```
calculate_costs([15, 12, 2, 10, 21],
               [5, 4, 5, 6, 3],
               [1, 2, 2, 3, 2],
               [5, 9, 1])
```

У цьому прикладі ми маємо $N = 5$ артефактів і $Q = 3$ запитів.

У першому запиті, $D = 5$. Ви можете відправити артефакти 0 і 3 в одному човні (оскільки $|15 - 10| \leq 5$) а решта артефактів в окремих човнах. Це дає мінімальну вартість транспортування всіх артефактів, яка становить $1 + 4 + 5 + 3 + 3 = 16$.

У другому запиті, $D = 9$. Ви можете відправити артефакти 0 і 1 в одному човні (оскільки $|15 - 12| \leq 9$) і відправити артефакти 2 і 3 в одному човні (оскільки $|2 - 10| \leq 9$). Останній артефакт можна відправляти окремим човном. Це дає мінімальну вартість транспортування всіх артефактів, яка становить $1 + 2 + 2 + 3 + 3 = 11$.

В останньому запиті, $D = 1$. Вам потрібно відправити кожен артефакт на своєму човні. Це дає мінімальну вартість транспортування всіх артефактів, яка становить $5 + 4 + 5 + 6 + 3 = 23$.

Отже, ця функція має повернути $[16, 11, 23]$.

Приклад градера

Формат вхідних даних:

```
N
W[0] A[0] B[0]
W[1] A[1] B[1]
...
W[N-1] A[N-1] B[N-1]
Q
E[0]
E[1]
...
E[Q-1]
```

Формат вихідних даних:

```
R[0]
R[1]
...
R[S-1]
```

Тут, S це довжина масиву R , який повертає `calculate_costs`.