

Nile

Вы хотите перевезти по Нилу N артефактов. Артефакты пронумерованы от 0 до $N - 1$. Вес артефакта i ($0 \leq i < N$) равен $W[i]$.

Чтобы перевезти артефакты вы используете специальные лодки. Каждая лодка может перевезти **не более двух** артефактов.

- Если вы перевозите один артефакт на лодке, его вес может быть любым.
- Если вы перевозите на лодке два артефакта, вам необходимо добиться того, чтобы лодка была отбалансирована. А именно, можно перевезти артефакты p и q ($0 \leq p < q < N$) на одной лодке, только если модуль разности между их весами не превышает D , то есть $|W[p] - W[q]| \leq D$.

Чтобы перевезти артефакт, вам необходимо заплатить стоимость, которая зависит от числа артефактов, которые перевозятся на этой лодке. Стоимость перевозки артефакта i ($0 \leq i < N$) равна:

- $A[i]$, если вы перевозите этот артефакт на отдельной лодке, либо
- $B[i]$, если вы перевозите на лодке два артефакта: этот и какой-либо еще.

Обратите внимание, что во втором случае вам необходимо оплатить перевозку обоих артефактов на этой лодке. А именно, если вы решите перевезти артефакты p и q ($0 \leq p < q < N$) на одной лодке, за это суммарно необходимо заплатить $B[p] + B[q]$.

Перевезти артефакт на отдельной лодке всегда дороже, чем перевезти его вместе с другим артефактом, так что $B[i] < A[i]$ для всех i , таких что $0 \leq i < N$.

К сожалению, река ведет себя очень непредсказуемо, поэтому значение D часто меняется. Ваша задача — ответить на Q запросов, пронумерованных от 0 до $Q - 1$. Запросы описаны массивом E длиной Q . Ответ на запрос j ($0 \leq j < Q$) — это минимальная стоимость перевозки всех N артефактов, если значение D равно $E[j]$.

Implementation Details

Вам необходимо реализовать следующую функцию:

```
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 : целочисленные массивы длины N , описывающие веса артефактов и стоимость их перевозки.
- E : целочисленный массив длины Q , задающий значения D для запросов.
- Функция должна вернуть массив R , содержащий Q целых чисел — минимальные стоимости перевозки артефактов, где $R[j]$ равно стоимости перевозки артефактов для запроса, в котором D равно $E[j]$ (для всех j , таких что $0 \leq j < Q$).
- Эта функция будет вызвана ровно один раз для каждого теста.

Constraints

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

Subtasks

Подзадача	Баллы	Дополнительные ограничения
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ для всех i , таких что $0 \leq i < N$
2	13	$Q \leq 5; W[i] = i + 1$ для всех i , таких что $0 \leq i < N$
3	17	$Q \leq 5; A[i] = 2$ и $B[i] = 1$ для всех i , таких что $0 \leq i < N$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ и $B[i] = 1$ для всех i , таких что $0 \leq i < N$
7	18	Нет дополнительных ограничений.

Example

Рассмотрим следующий вызов

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

В этом примере $N = 5$ артефактов и $Q = 3$ запроса.

В первом запросе $D = 5$. Вы можете перевезти артефакты 0 и 3 на одной лодке (так как $|15 - 10| \leq 5$), а остальные артефакты на отдельной лодке каждый. При этом получается минимальная стоимость перевозки всех артефактов, она равна $1 + 4 + 5 + 3 + 3 = 16$.

Во втором запросе $D = 9$. Вы можете перевезти артефакты 0 и 1 на одной лодке (так как $|15 - 12| \leq 9$), и перевезти артефакты 2 и 3 на одной лодке (так как $|2 - 10| \leq 9$). Оставшийся артефакт придется отправить на отдельной лодке. При этом получается минимальная стоимость перевозки всех артефактов, она равна $1 + 2 + 2 + 3 + 3 = 11$.

В последнем запросе $D = 1$. Необходимо отправить каждый артефакт на отдельной лодке. При этом получается минимальная стоимость перевозки всех артефактов, она равна $5 + 4 + 5 + 6 + 3 = 23$.

Таким образом функция должна вернуть $[16, 11, 23]$.

Sample Grader

Input format:

```
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]
```

Output format:

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

Здесь S это длина массива R , который вернул вызов `calculate_costs`.