

Нил

Искате да транспортирате N артефакта по реката Нил. Артефактите са номерирани от 0 до $N - 1$. Теглото на артефакт i ($0 \leq i < N$) е $W[i]$.

За да транспортирате артефактите, използвате специални лодки. Всяка лодка може да пренася **най-много два** артефакта.

- Ако решите да сложите само един артефакт в лодка, той може да е с произволно тегло, т.е. една лодка винаги може да носи всеки един артефакт, ако е сам.
- Ако решите да сложите два артефакта в една и съща лодка, трябва тя да е приблизително балансирана. По-конкретно, можете да пратите артефакти p и q в една и съща лодка, тогава и само тогава когато абсолютната стойност на разликата между техните тежести е по-малка или равна на D , т.е. $|W[p] - W[q]| \leq D$.

За да транспортирате някой артефакт, трябва да платите цена, която зависи от това дали е сам или не. Цената за транспортиране на артефакт i ($0 \leq i < N$) е:

- $A[i]$, ако го сложите сам в лодка, или
- $B[i]$, ако го сложите в лодка заедно с някой друг артефакт.

Забележете, че във втория случай трябва да платите цените и на двата артефакта в лодката. По-точно, ако решите да изпратите артефакти p и q ($0 \leq p < q < N$) в една и съща лодка, трябва да платите $B[p] + B[q]$.

Да изпратите артефакт сам в лодка винаги е по-скъпо от това да го изпратите заедно с някой друг артефакт, т.е. $B[i] < A[i]$ за всяко $0 \leq i < N$.

Искате да откриете минималната възможна обща цена за транспортиране на всички N артефакта точно по веднъж. За нещастие реката е много непредвидима и стойността на D се сменя често. Вашата задача е да отговорите на Q заявки, номерирани от 0 до $Q - 1$. Заявките се описват чрез списък от стойности E с дължина Q . Отговорът на заявка j ($0 \leq j < Q$) е търсената минимална цена, когато стойността на D е равна на $E[j]$.

Детайли по имплементацията

Трябва да имплементирате следната функция:

```
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 : вектори от цели числа с дължина N , описващи теглата на артефактите и цените за транспортирането им.
- E : вектор от цели числа с дължина Q , описващ стойността на D за всяка заявка.
- Функцията трябва да върне вектор R от Q цели числа, който да съдържа отговорите на заявките, т.е. $R[j]$ трябва да е минималната възможна обща цена за транспортиране на артефактите, когато D е равно на $E[j]$ (за всяко $0 \leq j < Q$).
- Тази функция се вика точно веднъж.

Ограничения

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ за всяко $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$ за всяко $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$ за всяко $0 \leq j < Q$

Подзадачи

Подзадача	Точки	Допълнителни ограничения
1	6	$Q \leq 5$; $N \leq 2000$; $W[i] = 1$ за всяко $0 \leq i < N$
2	13	$Q \leq 5$; $W[i] = i + 1$ за всяко $0 \leq i < N$
3	17	$Q \leq 5$; $A[i] = 2$ и $B[i] = 1$ за всяко $0 \leq i < N$
4	11	$Q \leq 5$; $N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ и $B[i] = 1$ за всяко $0 \leq i < N$
7	18	Няма.

Пример

Да разгледаме следното викане на функция:

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

В този пример има $N = 5$ артефакта и $Q = 3$ заявки.

В първата заявка $D = 5$. Можете да изпратите артефакти 0 и 3 в една лодка (защото $|15 - 10| \leq 5$), а останалите артефакти в отделни лодки. Това е решение с минимална цена за транспортиране на всички артефакти. Тази цена е: $1 + 4 + 5 + 3 + 3 = 16$.

Във втората заявка $D = 9$. Можете да изпратите артефакти 0 и 1 в една лодка (защото $|15 - 12| \leq 9$), артефакти 2 и 3 също в една (друга) лодка (защото $|2 - 10| \leq 9$), а останалите артефакти в отделни лодки. Това е решение с минимална цена: $1 + 2 + 2 + 3 + 3 = 11$.

В последната заявка $D = 1$. Тук трябва да изпратите всички артефакти в отделни лодки. Това е решение с минимална цена: $5 + 4 + 5 + 6 + 3 = 23$.

Следва, че функцията трябва да върне $[16, 11, 23]$.

Локален грейдър

Входен формат:

```
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]
```

Изходен формат:

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

Тук S е дължината на вектора R , върнат от `calculate_costs`.