

Nile

Bạn muốn vận chuyển N cổ vật qua sông Nile. Các cổ vật được đánh số từ 0 đến $N - 1$. Khối lượng của cổ vật i ($0 \leq i < N$) là $W[i]$.

Để vận chuyển các cổ vật, bạn dùng nhiều thuyền chuyên dụng. Mỗi chiếc thuyền có thể mang **nhiều nhất hai** cổ vật.

- Nếu bạn quyết định đặt duy nhất một cổ vật lên một chiếc thuyền, khối lượng của cổ vật có thể nặng tùy ý.
- Nếu bạn muốn đặt hai cổ vật trên cùng một chiếc thuyền, bạn phải chắc chắn rằng chiếc thuyền được cân bằng. Cụ thể, bạn có thể vận chuyển cổ vật p và q ($0 \leq p < q < N$) trên cùng một chiếc thuyền chỉ khi chênh lệch tuyệt đối giữa khối lượng của chúng nhiều nhất là D , nghĩa là $|W[p] - W[q]| \leq D$.

Để vận chuyển một cổ vật, bạn phải trả chi phí phụ thuộc vào số lượng cổ vật trên cùng một chiếc thuyền đó. Chi phí để vận chuyển cổ vật i ($0 \leq i < N$) là:

- $A[i]$, nếu bạn đặt duy nhất cổ vật này trên chiếc thuyền mang nó, hoặc
- $B[i]$, nếu bạn đặt cổ vật này trên thuyền cùng với một cổ vật khác.

Lưu ý rằng, trong trường hợp thứ hai, bạn phải trả chi phí cho cả hai cổ vật trên chiếc thuyền đó. Cụ thể, nếu bạn quyết định vận chuyển cổ vật p và q ($0 \leq p < q < N$) trên cùng một chiếc thuyền, bạn phải trả chi phí $B[p] + B[q]$.

Vận chuyển duy nhất một cổ vật trên một chiếc thuyền luôn luôn có chi phí đắt hơn vận chuyển cổ vật đó cùng với một cổ vật khác trên thuyền, tức là $B[i] < A[i]$ với mọi i thoả mãn $0 \leq i < N$.

Không may, dòng sông rất khó dự đoán và giá trị D thay đổi thường xuyên. Nhiệm vụ của bạn là trả lời Q câu hỏi, đánh số từ 0 đến $Q - 1$. Các câu hỏi được mô tả bằng một mảng E có độ dài Q . Câu trả lời cho câu hỏi j ($0 \leq j < Q$) là tổng chi phí nhỏ nhất để vận chuyển tất cả N cổ vật khi giá trị D bằng $E[j]$.

Chi tiết cài đặt

Bạn cần cài đặt hàm sau.

```
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 : các mảng số nguyên có độ dài N , mô tả khối lượng của các cổ vật và chi phí vận chuyển chúng.
- E : mảng số nguyên có độ dài Q mô tả giá trị của D cho từng câu hỏi.
- Hàm này cần trả về mảng R gồm Q số nguyên chứa tổng chi phí nhỏ nhất để vận chuyển các cổ vật, trong đó $R[j]$ là chi phí khi giá trị của D là $E[j]$ (với mỗi j thoả mãn $0 \leq j < Q$).
- Hàm này được gọi đúng một lần cho mỗi trường hợp test.

Các ràng buộc

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ với mọi i thoả mãn $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$ với mọi i thoả mãn $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$ với mọi j thoả mãn $0 \leq j < Q$

Các subtask

Subtask	Điểm	Các ràng buộc thêm
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ với mọi i thoả mãn $0 \leq i < N$
2	13	$Q \leq 5; W[i] = i + 1$ với mọi i thoả mãn $0 \leq i < N$
3	17	$Q \leq 5; A[i] = 2$ và $B[i] = 1$ với mọi i thoả mãn $0 \leq i < N$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ và $B[i] = 1$ với mọi i thoả mãn $0 \leq i < N$
7	18	Không có ràng buộc nào thêm.

Ví dụ

Xét lời gọi hàm sau.

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

Trong ví dụ này, ta có $N = 5$ cổ vật và $Q = 3$ câu hỏi.

Trong câu hỏi thứ nhất, $D = 5$. Bạn có thể vận chuyển cổ vật 0 and 3 trên cùng một chiếc thuyền (vì $|15 - 10| \leq 5$) và cổ vật còn lại có thể được vận chuyển trên các chiếc thuyền riêng biệt. Làm như vậy được tổng chi phí nhỏ nhất để vận chuyển tất cả cổ vật là $1 + 4 + 5 + 3 + 3 = 16$.

Trong câu hỏi thứ hai, $D = 9$. Bạn có thể vận chuyển cổ vật 0 and 1 trên cùng một chiếc thuyền (vì $|15 - 12| \leq 9$) và vận chuyển cổ vật 2 và 3 trên cùng một chiếc thuyền (vì $|2 - 10| \leq 9$). Cổ vật còn lại có thể được vận chuyển trên một chiếc thuyền riêng biệt. Làm như vậy được tổng chi phí nhỏ nhất để vận chuyển tất cả cổ vật là $1 + 2 + 2 + 3 + 3 = 11$.

Trong câu hỏi cuối cùng, $D = 1$. Bạn cần vận chuyển từng cổ vật trên một chiếc thuyền riêng biệt. Làm như vậy được tổng chi phí nhỏ nhất để vận chuyển tất cả cổ vật là $5 + 4 + 5 + 6 + 3 = 23$.

Do đó, hàm cần trả về $[16, 11, 23]$.

Trình chấm mẫu

Định dạng dữ liệu vào:

```
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]
```

Định dạng kết quả ra:

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

Trong đó, S là độ dài của mảng R trả về bởi hàm `calculate_costs`.