

Nil

Chcesz przetransportować N artefaktów przez Nil. Artefakty są ponumerowane od 0 do $N - 1$. Waga i -tego artefaktu ($0 \leq i < N$) jest równa $W[i]$.

Do transportu artefaktów używa się specjalistycznych łodzi. Każda łódź może przewieźć **co najwyżej dwa** artefakty.

- Jeśli chcesz umieścić w łodzi tylko jeden artefakt, jego waga może być dowolna.
- Jeśli chcesz umieścić dwa artefakty w tej samej łodzi, musisz upewnić się, że łódź jest równomiernie wyważona. Konkretnie, możesz wysłać artefakty p i q ($0 \leq p < q < N$) w tej samej łodzi tylko wtedy, gdy bezwzględna różnica ich wag wynosi co najwyżej D , czyli $|W[p] - W[q]| \leq D$.

Aby przetransportować artefakt, musisz zapłacić koszt zależny od liczby artefaktów przewożonych w tej samej łodzi. Koszt transportu i -tego artefaktu ($0 \leq i < N$) wynosi:

- $A[i]$, jeśli umieścisz tylko ten artefakt w łodzi, lub
- $B[i]$, jeżeli umieścisz go w łodzi razem z innym artefaktem.

Należy pamiętać, że w drugim przypadku trzeba zapłacić za oba artefakty znajdujące się w łodzi. Konkretnie, jeśli zdecydujesz się wysłać artefakty p i q ($0 \leq p < q < N$) w tej samej łodzi, musisz zapłacić $B[p] + B[q]$.

Wysyłka pojedynczego artefaktu łodzią zawsze jest droższa, niż gdyby wysłać go z jakimś innym artefaktem, to znaczy zachodzi $B[i] < A[i]$, dla każdego i ($0 \leq i < N$).

Niestety rzeka jest niepokorna i wartość D często się zmienia. Twoim zadaniem jest odpowiedzieć na Q zapytań ponumerowanych od 0 do $Q - 1$. Zapytania opisane są za pomocą tablicy E o długości Q . Odpowiedzią na j -te zapytanie ($0 \leq j < Q$) jest minimalny całkowity koszt transportu wszystkich N artefaktów dla wartości D równej $E[j]$.

Szczegóły implementacji

Powinieneś zaimplementować poniższą funkcję.

```
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 : tablice liczb całkowitych o długości N , opisujące wagę artefaktów i koszty ich transportu.
- E : tablica liczb całkowitych o długości Q , opisująca wartość D dla kolejnych zapytań.
- Ta funkcja powinna zwrócić tablicę R zawierającą Q liczb całkowitych, będących minimalnym całkowitym kosztem transportu artefaktów, gdzie $R[j]$ powinno być odpowiedzią, gdy wartość D wynosi $E[j]$ (dla j takiego, że $0 \leq j < Q$).
- Ta funkcja jest wywoływana dokładnie raz dla każdego testu.

Ograniczenia

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ dla każdego i takiego, że $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$ dla każdego i takiego, że $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$ dla każdego j takiego, że $0 \leq j < Q$

Podzadania

Podzadanie	Punkty	Dodatkowe ograniczenia
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ dla każdego i takiego, że $0 \leq i < N$
2	13	$Q \leq 5; W[i] = i + 1$ dla każdego i takiego, że $0 \leq i < N$
3	17	$Q \leq 5; A[i] = 2$ oraz $B[i] = 1$ dla każdego i takiego, że $0 \leq i < N$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ i $B[i] = 1$ dla każdego i takiego, że $0 \leq i < N$
7	18	brak dodatkowych ograniczeń

Przykład

Rozważmy poniższe wywołanie.

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

W tym przykładzie mamy $N = 5$ artefaktów i $Q = 3$ zapytań.

W pierwszym zapytaniu $D = 5$. Artefakty 0 i 3 możesz wysłać w jednej łodzi (ponieważ $|15 - 10| \leq 5$), a pozostałe artefakty oddzielnymi łodziami. Daje to minimalny koszt transportu

wszystkich artefaktów, który wynosi $1 + 4 + 5 + 3 + 3 = 16$.

W drugim zapytaniu $D = 9$. Możesz wysłać artefakty 0 i 1 w jednej łodzi (ponieważ $|15 - 12| \leq 9$) oraz wysłać artefakty 2 i 3 w drugiej łodzi (ponieważ $|2 - 10| \leq 9$). Pozostały artefakt możesz wysłać osobną łodzią. Daje to minimalny koszt transportu wszystkich artefaktów, który wynosi $1 + 2 + 2 + 3 + 3 = 11$.

W ostatnim zapytaniu $D = 1$. Musisz wysłać każdy artefakt w osobnej łodzi. Daje to minimalny koszt transportu wszystkich artefaktów, który wynosi $5 + 4 + 5 + 6 + 3 = 23$.

Dlatego takie wywołanie powinno zwrócić $[16, 11, 23]$.

Przykładowy program oceniający

Format wejścia:

```
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]
```

Format wyjścia:

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

S jest długością tablicy R zwracanej przez `calculate_costs`.