Nile

Vuoi trasportare N reperti attraverso il Nilo, numerati da 0 a $N-1$. Il peso del reperto i ($0 \leq i < N$) è $W[i]$.

Puoi trasportare i reperti tramite apposite imbarcazioni, che possono trasportare ciascuna **al massimo due** reperti.

- Se decidi di mettere un singolo reperto su una barca, può avere peso qualsiasi.
- Se vuoi mettere due reperti sulla stessa barca, devi assicurarti che la barca sia bilanciata. Nello specifico, puoi trasportare i reperti p e q ($0 \leq p < q < N$) nella stessa barca se e solo se la differenza assoluta tra i loro pesi è al massimo D, ovvero $|W[p] - W[q]| \leq D$.

Per trasportare un reperto, devi pagare un costo che dipende dal numero di reperti trasportati sulla sua imbarcazione. Il costo per trasportare il reperto i ($0 \leq i < N$) è:

- $A[i]$, se è l'unico reperto nella sua barca, o
- $B[i]$, se lo metti su una barca insieme a un altro reperto.

Tieni presente che in quest'ultimo caso dovrai pagare per entrambi i reperti presenti sulla barca. Nello specifico, se decidi di trasportare i reperti p e q ($0 \leq p < q < N$) nella stessa barca, devi pagare $B[p] + B[q]$.

Trasportare un reperto da solo è sempre più costoso che trasportarlo insieme a qualche altro reperto, cioè $B[i] < A[i]$ per tutti gli $0 \leq i < N$.

Sfortunatamente, il fiume è imprevedibile e il valore di D cambia spesso. Devi quindi rispondere a Q query, numerate da 0 a $Q-1$, descritte da un array E di lunghezza Q. La risposta alla domanda j ($0 \leq j < Q$) è il costo totale minimo del trasporto di tutti gli N reperti, se il valore di D è uguale a $E[j]$.

Note di implementazione

Devi implementare la seguente funzione.

```
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: array di interi di lunghezza N, che descrivono i pesi dei reperti e i costi del loro trasporto.
- E: un array di interi di lunghezza Q che descrive il valore di D per ogni domanda.
- La funzione deve restituire un array R di Q interi contenente il costo totale minimo del trasporto dei reperti, dove $R[j]$ è il costo se il valore di D è $E[j]$ (per ogni $0 \leq j < Q$).
- Questa funzione viene chiamata esattamente una volta per ogni caso di test.

Assunzioni

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq W[i] \leq 10^9$ per ogni i tale che $0 \leq i < N$
- $1 \leq B[i] < A[i] \leq 10^9$ per ogni i tale che $0 \leq i < N$
- $1 \leq E[j] \leq 10^9$ per ogni j tale che $0 \leq j < Q$

Subtask

Subtask	Punteggio	Limitazioni aggiuntive
1	6	$Q \leq 5$; $N \leq 2000$; $W[i] = 1$ per ogni $0 \leq i < N$
2	13	$Q \leq 5$; $W[i] = i + 1$ per ogni $0 \leq i < N$
3	17	$Q \leq 5$; $A[i] = 2$ e $B[i] = 1$ per ogni $0 \leq i < N$
4	11	$Q \leq 5$; $N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ e $B[i] = 1$ per ogni $0 \leq i < N$
7	18	Nessuna limitazione aggiuntiva.

Esempio

Consideriamo la seguente chiamata.

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

In questo esempio abbiamo $N = 5$ reperti e $Q = 3$ query.

Nella prima query $D = 5$. Puoi trasportare i reperti 0 e 3 nella stessa barca (poiché $|15 - 10| \leq 5$) e i reperti rimanenti in barche separate. Questo realizza il costo minimo per il trasporto di tutti i reperti, pari a $1 + 4 + 5 + 3 + 3 = 16$.

Nella seconda query $D = 9$. Puoi trasportare i reperti 0 e 1 nella stessa barca (poiché $|15 - 12| \leq 9$); i reperti 2 e 3 in una stessa barca (poiché $|2 - 10| \leq 9$); e il reperto rimanente in un'imbarcazione a parte. Questo realizza il costo minimo per il trasporto di tutti i reperti, pari a $1 + 2 + 2 + 3 + 3 = 11$.

Nell'ultima query $D = 1$. Devi trasportare ogni reperto in un barca diversa. Questo realizza il costo minimo per il trasporto di tutti i reperti, pari a $5 + 4 + 5 + 6 + 3 = 23$.

Pertanto, la funzione deve restituire $[16, 11, 23]$.

Grader di esempio

Formato di input:

```
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]
```

Formato di output:

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

dove S è la lunghezza dell'array R restituito da `calculate_costs`.