

## Nil

Du möchtest  $N$  Statuen über den Nil transportieren. Die Statuen sind von  $0$  bis  $N - 1$  nummeriert. Statue  $i$  ( $0 \leq i < N$ ) hat das Gewicht  $W[i]$ .

Für den Transport der Statuen kannst du spezielle Boote einsetzen. Mit jedem dieser Boote kannst du **höchstens zwei** Statuen transportieren:

- Wenn du auf einem Boot genau eine Statue transportierst, kann diese beliebig viel wiegen.
- Wenn du auf einem Boot zwei Statuen transportierst, musst du auf die Balance des Bootes achten. Das heißt: Du kannst zwei Statuen  $p$  und  $q$  ( $0 \leq p < q < N$ ) nur dann auf demselben Boot transportieren, wenn sich deren Gewichte um höchstens  $D$  unterscheiden:  $|W[p] - W[q]| \leq D$ .

Der Transport einer Statue verursacht Kosten. Diese Kosten hängen davon ab, ob die Statue einzeln auf einem Boot transportiert wird oder nicht. Die Kosten für den Transport von Statue  $i$  ( $0 \leq i < N$ ) betragen konkret:

- $A[i]$ , wenn du die Statue einzeln auf einem Boot transportierst;
- $B[i]$ , wenn du sie gemeinsam mit einer anderen Statue transportierst.

Im zweiten Fall hast du Kosten für beide Statuen  $p$  und  $q$  im Boot, also insgesamt  $B[p] + B[q]$ .

Verständlicherweise ist es immer teurer, eine Statue einzeln zu transportieren als gemeinsam mit einer anderen; es gilt also  $B[i] < A[i]$  ( $0 \leq i < N$ ).

Die Bedingungen auf dem Nil schwanken ständig, und so ändert sich auch der maximale Gewichtsunterschied  $D$  häufig. Du willst gerne für viele Werte von  $D$  im Voraus wissen, wie hoch die Gesamtkosten im besten Fall sein können, also bei optimaler Verteilung der Statuen auf Boote. Deshalb willst du  $Q$  Fragen beantworten, nummeriert von  $0$  bis  $Q - 1$ . Die Fragen sind durch ein Array  $E$  der Länge  $Q$  bestimmt: Die Antwort auf Frage  $j$  ( $0 \leq j < Q$ ) sind die minimalen Gesamtkosten für den Transport aller  $N$  Statuen für den Fall  $D = E[j]$ .

## Angaben zur Implementierung

Du sollst die folgende Funktion implementieren:

```
std::vector<long long> calculate_costs(
    std::vector<int> W, std::vector<int> A,
    std::vector<int> B, std::vector<int> E)
```

- $W$ ,  $A$  und  $B$  sind Arrays, die jeweils  $N$  ganze Zahlen enthalten.  $W$  beschreibt die Gewichte der Statuen,  $A$  und  $B$  die Kosten für den Transport der Statuen.
- $E$  ist ein Array, das  $Q$  ganze Zahlen enthält: die jeweiligen Werte von  $D$  für die einzelnen Fragen.
- Diese Funktion soll ein Array  $R$  von  $Q$  ganzen Zahlen zurückgeben.  $R$  soll für jede Frage die minimalen Gesamtkosten für den Transport der Statuen enthalten, wobei  $R[j]$  ( $0 \leq j < Q$ ) die Gesamtkosten angibt, wenn  $D$  den Wert  $E[j]$  hat.
- Für jeden Testfall wird diese Funktion genau einmal aufgerufen.

## Beschränkungen

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

## Subtasks

Subtask	Punkte	Zusätzliche Beschränkungen
1	6	$Q \leq 5$ ; $N \leq 2000$ ; $W[i] = 1$ für alle $i$ mit $0 \leq i < N$
2	13	$Q \leq 5$ ; $W[i] = i + 1$ für alle $i$ mit $0 \leq i < N$
3	17	$Q \leq 5$ ; $A[i] = 2$ und $B[i] = 1$ für alle $i$ mit $0 \leq i < N$
4	11	$Q \leq 5$ ; $N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ und $B[i] = 1$ für alle $i$ mit $0 \leq i < N$
7	18	Keine weiteren Beschränkungen.

## Beispiel

Im folgenden Aufruf gibt es  $N = 5$  Statuen und  $Q = 3$  Fragen.

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

Für die erste Frage ist  $D = 5$ . Du kannst Statuen 0 und 3 auf demselben Boot transportieren (weil  $|15 - 10| \leq 5$ ) und die verbleibenden Statuen jeweils einzeln. Das führt zu minimalen Gesamtkosten von  $1 + 4 + 5 + 3 + 3 = 16$ .

Für die zweite Frage ist  $D = 9$ . Du kannst jeweils Statuen 0 und 1 (weil  $|15 - 12| \leq 9$ ) und Statuen 2 und 3 (weil  $|2 - 10| \leq 9$ ) auf demselben Boot transportieren. Die verbleibende Statue kann einzeln transportiert werden. Das führt zu minimalen Gesamtkosten von  $1 + 2 + 2 + 3 + 3 = 11$ .

Für die letzte Frage ist  $D = 1$ . Hier musst du alle Statuen einzeln transportieren. Das führt zu minimalen Gesamtkosten von  $5 + 4 + 5 + 6 + 3 = 23$ .

Insgesamt soll die Funktion also  $[16, 11, 23]$  zurückgeben.

## Beispielgrader

Eingabeformat:

```
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]
```

Ausgabeformat:

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

Hier ist  $S$  die Länge des Arrays  $R$ , das `calculate_costs` zurückgegeben hat.