

Nil

Anda ingin mengantar N artefak melewati Nil. Artefak dinomori dari 0 hingga $N - 1$. Berat dari artefak i ($0 \leq i < N$) adalah $W[i]$.

Untuk mengantarkan artefak, Anda menggunakan kapal yang khusus. Tiap kapal bisa mengangkut **paling banyak dua** artefak.

- Jika Anda memutuskan untuk menaruh satu artefak di suatu kapal, berat artefaknya boleh bebas.
- Jika Anda ingin menaruh dua artefak di kapal yang sama, Anda harus memastikan kapal tersebut seimbang. Khususnya, Anda bisa mengirim artefak p dan q ($0 \leq p < q < N$) di kapal yang sama jika selisih mutlak dari berat mereka paling besar D , yaitu $|W[p] - W[q]| \leq D$.

Untuk mengangkut sebuah artefak, Anda perlu membayar biaya yang bergantung dengan banyaknya artefak yang diangkut oleh kapal yang sama. Biaya dari pengangkutan artefak i ($0 \leq i < N$) adalah:

- $A[i]$, jika Anda hanya menaruh artefak tersebut dengan sendirinya di suatu kapal, atau
- $B[i]$, jika Anda menaruh artefak tersebut bersama dengan artefak lainnya.

Catat bahwa untuk kasus yang kedua, Anda perlu membayar untuk kedua artefak di kapal tersebut. Khususnya, jika Anda memutuskan untuk mengirim artefak p dan q ($0 \leq p < q < N$) di kapal yang sama, Anda perlu membayar $B[p] + B[q]$.

Mengirim sebuah artefak di suatu kapal dengan sendirinya selalu lebih mahal dibanding mengirimnya bersama dengan artefak lainnya di kapal yang sama, sehingga $B[i] < A[i]$ untuk semua i sehingga $0 \leq i < N$.

Sayangnya, sungainya sangat tidak stabil dan nilai D berubah-ubah. Tugas Anda adalah menjawab Q pertanyaan yang dinomori dari 0 hingga $Q - 1$. Pertanyaan dideskripsikan dengan *array* E sepanjang Q . Jawaban dari pertanyaan j ($0 \leq j < Q$) adalah biaya total minimum dari pengangkutan semua N artefak ketika nilai D sama dengan $E[j]$.

Detail Implementasi

Anda harus mengimplementasikan prosedur berikut.

```
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 bilangan bulat sepanjang N yang mendeskripsikan berat dari artefak dan biaya pengangkutan mereka.
- E : array bilangan bulat sepanjang Q yang mendeskripsikan nilai dari D untuk setiap pertanyaan.
- Prosedur ini harus mengembalikan sebuah array R dengan Q bilangan bulat yang mengandung biaya total minimum dari pengangkutan artefak, dengan $R[j]$ adalah biaya ketika nilai D adalah $E[j]$ (untuk setiap j sehingga $0 \leq j < Q$).
- Prosedur ini dipanggil tepat sekali untuk setiap kasus uji.

Batasan

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

Batasan

Subtask	Nilai	Batasan tambahan
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ untuk setiap i sehingga $0 \leq i < N$
2	13	$Q \leq 5; W[i] = i + 1$ untuk setiap i sehingga $0 \leq i < N$
3	17	$Q \leq 5; A[i] = 2$ dan $B[i] = 1$ untuk setiap i sehingga $0 \leq i < N$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ dan $B[i] = 1$ untuk setiap i sehingga $0 \leq i < N$
7	18	Tidak ada batasan tambahan.

Contoh

Contoh 1

Perhatikan pemanggilan berikut.

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

Pada contoh ini, kita mempunyai $N = 5$ artefak dan $Q = 3$ pertanyaan.

Pada pertanyaan pertama, $D = 5$. Anda bisa mengirim artefak 0 dan 3 di satu kapal (karena $|15 - 10| \leq 5$) dan sisa artefaknya di kapal-kapal yang terpisah. Ini memberikan biaya minimum untuk mengangkut semua artefak, yaitu $1 + 4 + 5 + 3 + 3 = 16$.

Pada pertanyaan kedua, $D = 9$. Anda bisa mengirim artefak 0 dan 1 di satu kapal (karena $|15 - 12| \leq 9$) dan mengirim artefak 2 dan 3 di satu kapal (karena $|2 - 10| \leq 9$). Artefak yang tersisa dikirim menggunakan kapal yang terpisah. Ini memberikan biaya minimum untuk mengangkut semua artefak, yaitu $1 + 2 + 2 + 3 + 3 = 11$.

Pada pertanyaan terakhir, $D = 1$. Anda bisa mengirimkan tiap artefak di kapalnya tersendiri. Ini memberikan biaya minimum untuk mengangkut semua artefak, yaitu $5 + 4 + 5 + 6 + 3 = 23$.

Oleh karena itu, prosedur harus mengembalikan $[16, 11, 23]$.

Contoh Grader

Format masukan:

```
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 keluaran:

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

Di sini, S adalah panjang dari *array* R yang dikembalikan `calculate_costs`.