

Nīla

Jūs vēlaties pārvietot pa Nīlu N artefaktus. Artefakti ir sanumurēti ar skaitļiem no 0 līdz $N - 1$. Artefakta i ($0 \leq i < N$) svars ir $W[i]$.

Artefaktu transportēšanai tiek izmantotas specializētas laivas. Katrā laivā var iekraut **ne vairāk kā divus** artefaktus.

- Ja jūs izlemsiet laivā iekraut tikai vienu artefaktu, tā svars var būt patvaļīgs.
- Ja jūs izlemsiet vienā laivā iekraut divus artefaktus, jums jānodrošina, ka laiva ir pienācīgi līdzsvarota. Precīzāk, jūs varat vienā laivā iekraut artefaktus p un q ($0 \leq p < q < N$) tikai tad, ja to svaru absolūtā starpība nepārsniedz D , tas ir $|W[p] - W[q]| \leq D$.

Artefakta transportēšana izmaksā noteiktu naudas summu, kas ir atkarīga no artefaktu skaita, kas tiek transportēti vienā laivā. Artefakta i ($0 \leq i < N$) transportēšanas izmaksas ir:

- $A[i]$, ja artefakts tiek transportēts laivā viens pats, vai
- $B[i]$, ja tas laivā tiek transportēts kopā ar kādu citu artefaktu.

Ievērojiet, ka pēdējā gadījumā jums būs jāmaksā par abiem laivā esošajiem artefaktiem. Precīzāk, ja jūs nolemsiet vienā laivā transportēt artefaktus p un q ($0 \leq p < q < N$), jums nāksies samaksāt $B[p] + B[q]$.

Transportēt artefaktu laivā vienu pašu vienmēr ir dārgāk, nekā transportēt to kopā ar kādu citu, tāpēc $B[i] < A[i]$ visiem i , kur $0 \leq i < N$.

Diemžēl, upe ir ļoti neparedzama un D vērtība bieži mainās. Jums jāatbild uz Q jautājumiem, kas sanumurēti no 0 līdz $Q - 1$. Jautājumi ir aprakstīti masīvā E , kura garums ir Q . Atbilde uz j -to jautājumu ($0 \leq j < Q$) ir mazākās kopējās N artefaktu transportēšanas izmaksas, ja D vērtība ir $E[j]$.

Implementēšanas detaļas

Jums jāimplementē šāda procedūra:

```
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 : naturālu skaitļu masīvi garumā N , kas apraksta artefaktu svarus un to transportēšanas izmaksas.
- E : naturālu skaitļu masīvs garumā Q , kas apraksta D vērtības dažādiem jautājumiem.
- Procedūrai jāatgriež masīvs R no Q naturāliem skaitļiem, kur katrs skaitlis ir mazākās kopējās visu artefaktu transportēšanas izmaksas, kur $R[j]$ ir kopējās izmaksas, kad D vērtība ir $E[j]$ (visiem j , kur $0 \leq j < Q$).
- Šī procedūra katram testam tiks izsaukta vienreiz.

Ierobežojumi

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

Apakšuzdevumi

Apakšuzdevums	Punkti	Papildu ierobežojumi
1	6	$Q \leq 5$; $N \leq 2000$; $W[i] = 1$ visiem i , kur $0 \leq i < N$
2	13	$Q \leq 5$; $W[i] = i + 1$ visiem i , kur $0 \leq i < N$
3	17	$Q \leq 5$; $A[i] = 2$ un $B[i] = 1$ visiem i , kur $0 \leq i < N$
4	11	$Q \leq 5$; $N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ un $B[i] = 1$ visiem i , kur $0 \leq i < N$
7	18	Bez papildu ierobežojumiem.

Piemērs

Aplūkosim šādu izsaukumu:

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

Šajā piemērā ir $N = 5$ artefakti un $Q = 3$ jautājumi.

Pirmajā jautājumā $D = 5$. Jūs varat vienā laivā transportēt artefaktus 0 un 3, jo $|15 - 10| \leq 5$, un atlikušos artefaktus atsevišķās laivās. Tas nozīmē, ka mazākās kopējās artefaktu transportēšanas

izmaksas ir $1 + 4 + 5 + 3 + 3 = 16$.

Otrajā jautājumā $D = 9$. Jūs varat vienā laivā transportēt artefaktus 0 un 1, jo $|15 - 12| \leq 9$, kā arī artefaktus 2 un 3, jo $|2 - 10| \leq 9$. Atlikušais artefakts jātransportē atsevišķā laivā. Tas nozīmē, ka mazākās kopējās artefaktu transportēšanas izmaksas ir $1 + 2 + 2 + 3 + 3 = 11$.

Un, visbeidzot, jautājums, kur $D = 1$. Visi artefakti jātransportē atsevišķās laivās. Tas nozīmē, ka mazākās kopējās artefaktu transportēšanas izmaksas ir $5 + 4 + 5 + 6 + 3 = 23$.

Tādējādi, procedūrai jāatgriež masīvs $[16, 11, 23]$.

Paraugvērtētājs

Ievaddatu formāts:

```
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]
```

Izvaddatu formāts:

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

Šeit ar S apzīmēts masīva R , kuru atgriež `calculate_costs`, garums.