

# Nijl

Je wil  $N$  artefacten over de Nijl vervoeren. De artefacten zijn genummerd van 0 tot en met  $N - 1$ . Het gewicht van artefact  $i$  ( $0 \leq i < N$ ) is  $W[i]$ .

Om de artefacten te vervoeren, kun je gespecialiseerde boten gebruiken. Elke boot kan **maximaal twee** artefacten vervoeren.

- Als je een enkel artefact in een boot stopt, kan het gewicht arbitrair zijn.
- Als je twee artefacten in dezelfde boot stopt, moet je ervoor zorgen dat de boot in balans is. Om precies te zijn, je kan artefacten  $p$  and  $q$  ( $0 \leq p < q < N$ ) in dezelfde boot vervoeren, alleen als het absolute verschil tussen de gewichten maximaal  $D$  is, dus  $|W[p] - W[q]| \leq D$ .

Om een artefact te vervoeren, moet je een hoeveelheid betalen die afhankelijk is van het aantal artefacten wat je op de boot meeneemt. De prijs voor het vervoeren van artefact  $i$  ( $0 \leq i < N$ ) is:

- $A[i]$ , als je het artefact alleen op een boot vervoert, of
- $B[i]$ , als je het artefact samen met een ander artefact op een boot vervoert.

Let erop dat je in het laatste geval, voor beide artefacten op de boot moet betalen. Om precies te zijn, als je besluit om artefacten  $p$  en  $q$  ( $0 \leq p < q < N$ ) op dezelfde boot te vervoeren, moet je  $B[p] + B[q]$  betalen.

Een artefact alleen op een boot versturen is altijd duurder dan het artefact samen met een ander artefact versturen, dus  $B[i] < A[i]$  voor alle  $i$  ( $0 \leq i < N$ ).

Helaas is de rivier heel onvoorspelbaar en verandert de waarde van  $D$  vaak. Jouw taak is om  $Q$  vragen te beantwoorden van 0 tot en met  $Q - 1$ . De vragen zijn geschreven in een array  $E$  met lengte  $Q$ . Het antwoord op vraag  $j$  ( $0 \leq j < Q$ ) is de minimale totale prijs voor het vervoeren van alle  $N$  artefacten, als de waarde van  $D$  gelijk is aan  $E[j]$ .

## Implementatie Details

Je moet de volgende procedure implementeren.

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

- $W, A, B$ : arrays van integers met lengte  $N$ , die de gewichten en de prijzen voor het vervoeren beschrijven.
- $E$ : een array van integers met lengte  $Q$ , die de waarden van  $D$  beschrijft voor elke vraag.
- Deze procedure moet een array  $R$  met  $Q$  integers returnen die de minimale prijs bevat voor het vervoeren van de artefacten, waar  $R[j]$  de prijs is wanneer de waarde van  $D$  gelijk is aan  $E[j]$  (for elke  $j$  zodat  $0 \leq j < Q$ ).
- Deze procedure is exact één keer aangeroepen voor elke test case.

## Beperkingen

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

## Subtaken

Subtaak	Score	Extra beperkingen
1	6	$Q \leq 5; N \leq 2000; W[i] = 1$ voor elke $i$ zodat $0 \leq i < N$
2	13	$Q \leq 5; W[i] = i + 1$ voor elke $i$ zodat $0 \leq i < N$
3	17	$Q \leq 5; A[i] = 2$ en $B[i] = 1$ voor elke $i$ zodat $0 \leq i < N$
4	11	$Q \leq 5; N \leq 2000$
5	20	$Q \leq 5$
6	15	$A[i] = 2$ en $B[i] = 1$ voor elke $i$ zodat $0 \leq i < N$
7	18	Geen extra beperkingen.

## Voorbeeld

Bekijk de volgende call.

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

In dit voorbeeld hebben we  $N = 5$  artefacten en  $Q = 3$  vragen.

In de eerste vraag is  $D = 5$ . Je kunt artefacten 0 en 3 in één boot sturen (aangezien  $|15 - 10| \leq 5$ ) en de overige artefacten in aparte boten. Dit levert de minimale prijs op voor het transporteren

van alle artefacten, namelijk  $1 + 4 + 5 + 3 + 3 = 16$ .

In de tweede vraag is  $D = 9$ . Je kunt artefacten 0 en 1 in één boot versturen (aangezien  $|15 - 12| \leq 9$ ) en artefacten 2 en 3 in één boot versturen (aangezien  $|2 - 10| \leq 9$ ). Het overige artefact kan in een aparte boot worden verzonden. Dit levert de minimale prijs op voor het transporteren van alle artefacten, namelijk  $1 + 2 + 2 + 3 + 3 = 11$ .

In de laatste vraag is  $D = 1$ . Je moet elk artefact in zijn eigen boot sturen. Dit levert de minimale prijs op voor het transporteren van alle artefacten, namelijk  $5 + 4 + 5 + 6 + 3 = 23$ .

Deze procedure zou dus  $[16, 11, 23]$  moeten returnen.

## Voorbeeldgrader

Invoerformaat:

```
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]
```

Uitvoerformaat:

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

Hier is  $S$  de lente van de array  $R$  die gereturt is door `calculate_costs`.