

## Parcel Post (multihop)


Om pakketten meer efficiënt te bezorgen heeft de Post Office een netwerk van pneumatische buizen gebouwd onder de straten van London. Het netwerk bestaat uit  $N$  sorteer-stations verbonden met  $N - 1$  ongerichte buizen. Er is een uniek pad tussen elk paar stations, en pakketten worden verstuurd langs het pad van hun startpunt naar hun eindpunt.

Als een pakket bij sorteerpunt  $i$  is, zijn er twee manieren om het richting het eindpunt te sturen. Het kan afgevuurd worden met *weinig kracht*, wat  $A_i$  kost en waardoor het via één buis verplaatst naar het volgende station langs de route. Anderzijds kan het met *veel kracht* afgevuurd worden. In dat geval kan de werknemer met een draaiknop een waarde  $k \geq 1$  selecteren, waarna het pakket door de volgende  $k$  buizen gaat, met een kost van  $B_i + k \cdot C$ .

De Post Office gebruikt de route die het minste kost, maar om verstopping van het netwerk te voorkomen moet elk pakket op de directe route van het startpunt naar het eindpunt blijven. Jouw opdracht is om de minimale kosten voor een reeks van  $Q$  pakketten, die met dit netwerk verzonden worden, uit te rekenen.

## Implementatie

Je moet een enkel `.cpp` bestand inleveren.

 In de bijlagen van deze opgave vind je een template `multihop.cpp` met een voorbeeld-implementatie.

Je moet de volgende functie implementeren:

C++	<code>void init(int N, int C, vector&lt;int&gt; A, vector&lt;int&gt; B, vector&lt;int&gt; U,</code>
	<code>vector&lt;int&gt; V);</code>
C++	<code>long long query(int X, int Y);</code>

- Integer  $N$  die staat voor het aantal sorteerstations.
- Integer  $C$  die staat voor de extra kosten per eenheid kracht wanneer er met veel kracht wordt afgevuurd, zoals hierboven beschreven.
- De lijst  $A$ , genummerd van 0 tot en met  $N - 1$ , die de kosten bevat om met weinig kracht af te vuren vanaf elk punt.
- De lijst  $B$ , genummerd van 0 tot en met  $N - 1$ , die de vaste kosten bevat om met veel kracht af te vuren vanaf elk punt, zoals hierboven beschreven.
- De lijsten  $U$  en  $V$  die de buizen in het netwerk beschrijven: Er is een buis tussen sorteerstation  $U[i]$  en sorteerstation  $V[i]$ .
- `query` moet de minimale kosten voor het versturen van een pakket van sorteerstation  $X$  naar sorteerstation  $Y$  teruggeven.

De grader roept de functie `init` aan, en zal daarna `query`  $Q$  keer aanroepen, waarbij de teruggegeven waarde wordt geprint in het uitvoerbestand.

## Voorbeeld Grader

De map van de opgave bevat een versimpelde versie van de jury's grader, die je kan gebruiken om je oplossing lokaal te testen. De versimpelde grader leest de invoer van `stdin`, roept de functies aan die je moet implementeren en schrijft uiteindelijk de uitvoer naar `stdout`.

De invoer bestaat uit  $N + Q + 2$  regels, die bevatten:

- Regel 1: de integers  $N, Q, C$ .
- Regel 2: de integers  $A_i$ , gescheiden door spaties.
- Regel 3: de integers  $B_i$ , gescheiden door spaties.
- Regel  $4 + i$  ( $0 \leq i < N - 1$ ): de integers  $U_i, V_i$ .
- Regel  $4 + (N - 1) + i$  ( $0 \leq i < Q$ ): de integers  $X_i, Y_i$ .

De uitvoer bestaat uit  $Q$  regels, de waarden die teruggegeven worden door de functie `query`.

## Randvoorwaarden

- $1 \leq N \leq 100\,000$ .
- $1 \leq Q \leq 100\,000$ .
- $1 \leq C \leq 1\,000\,000\,000$ .
- $1 \leq A_i \leq 1\,000\,000\,000$  voor alle  $i = 0, \dots, N - 1$
- $1 \leq B_i \leq 1\,000\,000\,000$  voor alle  $i = 0, \dots, N - 1$
- $0 \leq U_i < N$ .
- $0 \leq V_i < N$ .

## Scoring

Je programma wordt getest op een set testgevallen gegroepeerd per deelopgave. Om de score te krijgen die hoort bij een deelopgave, moet je alle testgevallen oplossen die daarbij horen.

- **Subtask 1** [ 0 punten]: Voorbeeld testgevallen.
- **Subtask 2** [ 5 punten]:  $A_i \leq 10, B_i \leq 10$  voor alle  $i = 0, \dots, N - 1, C \leq 10, N \leq 10, Q \leq 10$ .
- **Subtask 3** [10 punten]:  $N \leq 5000, Q = 1$ .
- **Subtask 4** [25 punten]:  $N \leq 100\,000, Q = 1$ .
- **Subtask 5** [25 punten]:  $N \leq 5000$ .
- **Subtask 6** [35 punten]: Geen extra randvoorwaarden.

## Voorbeelden

stdin	stdout
5 1 4 2 8 6 9 2 2 5 9 5 2 3 0 2 3 4 2 1 4 0 1	16
5 5 3 9 7 9 4 5 5 10 8 9 7 4 3 0 4 2 0 1 2 4 0 3 1 0 3 3 0 1 4	5 20 11 9 19

## Uitleg

In het **eerste voorbeeld**, kunnen we een pakket afvuren van sorteerstation 0 naar 4 met veel kracht, voor een kost van 14, daarna van 4 naar 1 met weinig kracht, voor een kost van 2. De totale kosten zijn 16, wat optimaal is.