

## Parcel Post (multihop)

Um Pakete effizienter auszuliefern, hat die Post ein Netzwerk von Rohrpostsystemen unter den Strassen von London gebaut. Dieses Netzwerk besteht aus  $N$  Verteilungsstationen, die durch  $N - 1$  ungerichtete Rohre miteinander verbunden sind. Zwischen jedem Paar von Stationen gibt es einen eindeutigen Pfad, und Pakete werden entlang dieses Pfads von ihrer Quelle zu ihrem Ziel gesendet.

Wenn ein Paket an der Verteilungsstation  $i$  ist, gibt es zwei Möglichkeiten, es auf den Weg zu seinem Ziel zu bringen. Es kann mit *geringer Leistung* zu Kosten von  $A_i$  abgefeuert werden, in diesem Fall reist es entlang eines einzelnen Rohrs zur nächsten Station auf seiner Route. Alternativ kann es mit *hoher Leistung* abgefeuert werden. In diesem Fall wird der Operator eine Schalteinstellung  $k \geq 1$  wählen, und das Paket wird entlang der nächsten  $k$  Rohre seiner Route reisen, zu Kosten von  $B_i + k \cdot C$ .

Die Post wird die Pakete so routen, dass die Gesamtkosten minimiert werden. Um jedoch Staus im Netzwerk zu vermeiden, müssen Pakete auf dem direkten Weg von ihrer Quelle zu ihrem Ziel bleiben. Deine Aufgabe ist es, die minimalen Kosten für eine Serie von  $Q$  Paketen zu finden, die in diesem Netzwerk gesendet werden sollen.

## Implementierung

Du musst eine einzelne .cpp-Quelldatei einreichen.

 Unter den Anhängen dieser Aufgabe findest Du eine Vorlage `multihop.cpp` mit einer Beispielimplementierung.

Du musst die folgenden Funktionen implementieren:

```
C++    void init(int N, int C, vector<int> A, vector<int> B, vector<int> U,
C++    vector<int> V);
C++    long long query(int X, int Y);
```

- Die Ganzzahl  $N$  repräsentiert die Anzahl der Verteilungsstationen.
- Die Ganzzahl  $C$  repräsentiert die Zusatzkosten pro Einheit Leistung beim Abfeuern mit hoher Leistung, wie oben beschrieben.
- Das Array  $A$ , indiziert von 0 bis  $N - 1$ , enthält die Kosten für das Abfeuern mit geringer Leistung von jedem Knoten aus.
- Das Array  $B$ , indiziert von 0 bis  $N - 1$ , enthält die Grundkosten für das Abfeuern mit hoher Leistung von jedem Knoten aus, wie oben beschrieben.
- Die Arrays  $U$  und  $V$  beschreiben die Rohre im Netzwerk: Es gibt ein Rohr zwischen der Verteilungsstation  $U[i]$  und der Verteilungsstation  $V[i]$ .
- `query` sollte die minimalen Kosten für das Senden eines Pakets von der Verteilungsstation  $X$  zur Verteilungsstation  $Y$  zurückgeben.

Der Grader wird die Funktion `init` aufrufen und danach `query`  $Q$  Mal aufrufen, um den Rückgabewert in die Ausgabedatei zu schreiben.

## Beispielgrader

Das Verzeichnis dieser Aufgabe enthält eine vereinfachte Version des Graders, die verwendet werden kann, um Deine Lösung lokal zu testen. Der vereinfachte Grader liest die Eingabedaten aus `stdin`, ruft die Funktionen auf, die Du implementieren musst, und schreibt schliesslich die Ausgabe in `stdout`.

Die Eingabe besteht aus  $N + Q + 2$  Zeilen, die folgendes enthalten:

- Zeile 1: die Ganzzahlen  $N, Q, C$ .
- Zeile 2: die Ganzzahlen  $A_i$ , getrennt durch Leerzeichen.
- Zeile 3: die Ganzzahlen  $B_i$ , getrennt durch Leerzeichen.
- Zeile  $4 + i$  ( $0 \leq i < N - 1$ ): die Ganzzahlen  $U_i, V_i$ .
- Zeile  $4 + (N - 1) + i$  ( $0 \leq i < Q$ ): die Ganzzahlen  $X_i, Y_i$ .

Die Ausgabe besteht aus  $Q$  Zeilen, die die von der Funktion `query` zurückgegebenen Werte enthalten.

## Einschränkungen

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

## Punktevergabe

Dein Programm wird an einer Reihe von Testfällen getestet, die in Subtasks gruppiert sind. Um die mit einem Subtask verbundene Punktzahl zu erhalten, musst Du alle darin enthaltenen Testfälle korrekt lösen.

- **Teilaufgabe 1** [ 0 Punkte]: Beispiel-Testfälle.
- **Teilaufgabe 2** [ 5 Punkte]:  $A_i \leq 10, B_i \leq 10$  für jedes  $i = 0, \dots, N - 1, C \leq 10, N \leq 10, Q \leq 10$ .
- **Teilaufgabe 3** [10 Punkte]:  $N \leq 5000, Q = 1$ .
- **Teilaufgabe 4** [25 Punkte]:  $N \leq 100\,000, Q = 1$ .
- **Teilaufgabe 5** [25 Punkte]:  $N \leq 5000$ .
- **Teilaufgabe 6** [35 Punkte]: Keine zusätzlichen Einschränkungen.

## Beispiele

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

## Erklärung

Im **ersten Beispiel** können wir das Paket von der Verteilungsstation 0 zur 4 mit hoher Leistung abschicken, zu Kosten von 14, und dann von 4 zur 1 mit geringer Leistung, zu Kosten von 2. Die endgültigen Kosten sind dann 16, was optimal ist.