

Parcel Post (multihop)

Afin de livrer des colis plus efficacement, La Poste a construit un réseau de tubes pneumatiques sous les rues de Londres. Le réseau consiste de N stations de routage reliées par $N - 1$ tubes bidirectionnels (utilisables dans les deux sens). Il existe un seul chemin entre chaque paire de stations et les colis seront envoyés le long de ces chemins, de la source à la destination.

Lorsqu'un colis arrive à la station de routage i , deux options d'envoi existent. Le colis peut être envoyé à *puissance faible* avec un coût A_i , il traversera alors un seul tube jusqu'à la prochaine station le long de la route. Alternativement, le colis peut être envoyé à *puissance élevée*. Dans ce cas, l'opérateur peut choisir un nombre $k \geq 1$ et le colis traversera les k prochains tubes le long de la route, avec un coût de $B_i + k \cdot C$.

La Poste va vouloir transporter les colis avec un coût total minimal, néanmoins afin d'éviter des embouteillages du réseau, les colis doivent suivre le chemin direct de la source vers leur destination. Votre tâche est de trouver les coûts minimaux d'une série de Q colis à envoyer dans le réseau.

Implémentation

Vous devez soumettre un seul fichier source `.cpp`.

🔍 Parmi les pièces jointes à cette tâche, vous trouverez un modèle `multihop.cpp` avec un modèle d'implémentation.

Vous devez implémenter les fonctions suivantes :

```
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);
```

- L'entier N représente le nombre de stations de routage.
- L'entier C représente le coût incrémental par unité de puissance lors d'un envoi à *puissance élevée*, comme décrit plus haut.
- Le tableau A , indexé de 0 à $N - 1$, contient les coûts d'un envoi à *puissance faible* à chaque station.
- Le tableau B , indexé de 0 à $N - 1$, contient les coûts de base d'un envoi à *puissance élevée* à chaque station, comme décrit plus haut.
- Les tableaux U et V décrivent les tubes du réseau : il y a un tube entre la station de routage $U[i]$ à la station $V[i]$.
- `query` doit retourner le coût minimal pour envoyer un colis de la station de routage X à la station Y .

Le grader appellera la fonction `init`, et ensuite `query` Q fois, imprimant les valeurs de retour dans le fichier de sortie.

Évaluateur

Le dossier de la tâche contient une version simplifiée du grader, qui vous permet de tester votre solutions localement. Ce grader simplifié lit les données d'entrée depuis `stdin`, appelle les fonctions que vous devez implémenter, et finalement écrit le résultat dans `stdout`

L'entrée contient $N + Q + 2$ lignes, de la façon suivante :

- Ligne 1 : les entiers N, Q, C .
- Ligne 2 : les entiers A_i , séparés par des espaces.
- Ligne 3 : les entiers B_i , séparés par des espaces.
- Ligne $4 + i$ ($0 \leq i < N - 1$) : les entiers U_i, V_i .
- Ligne $4 + (N - 1) + i$ ($0 \leq i < Q$) : les entiers X_i, Y_i .

La sortie est composée de Q lignes, contient les valeurs retournées par la fonction `query`.

Contraintes

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

Score

Votre programme sera testé sur un ensemble de cas de test groupés par sous-tâche. Afin d'obtenir les points associés à une sous-tâche, vous devez répondre correctement à tout les cas de test associés.

- **Sous-tâche 1** [**0 points**]: Les exemples.
- **Sous-tâche 2** [**5 points**]: $A_i \leq 10, B_i \leq 10$ pour chaque $i = 0, \dots, N - 1, C \leq 10, N \leq 10, Q \leq 10$.
- **Sous-tâche 3** [**10 points**]: $N \leq 5000, Q = 1$.
- **Sous-tâche 4** [**25 points**]: $N \leq 100\,000, Q = 1$.
- **Sous-tâche 5** [**25 points**]: $N \leq 5000$.
- **Sous-tâche 6** [**35 points**]: Pas de contraintes additionnelles.

Exemples

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

Explication

Dans le **premier exemple**, le colis peut être envoyé de la station de routage 0 à 4 avec une *puissance élevée*, pour un coût de 14, et ensuite de 4 à 1 avec une *puissance faible*, pour un coût de 2. Le coût final est donc de 16, ce qui est optimal.