

Parcel Post (multihop)

Para entregar paquetes de forma más eficiente, la *Post Office* ha construido una red de tuberías neumáticas bajo las calles de Londres. La red consiste en N estaciones de enrutado conectadas entre sí por $N - 1$ tuberías bidireccionales. Hay un único camino entre cada par de estaciones y los paquetes serán enviados por ese camino desde su origen hasta su destino.

Cuando un paquete está en la estación de enrutado i , hay dos opciones para enviarlo hacia su destino. Puede ser enviado a *baja potencia* a un coste A_i , en cuyo caso viajará por una única tubería hacia la siguiente estación de la ruta. Alternativamente, puede ser enviado a *alta potencia*. En este caso, el operador seleccionará un $k \geq 1$, y el paquete pasará por las siguientes k tuberías de su ruta, por un coste de $B_i + k \cdot C$.

La *Post Office* enviará los paquetes de forma que se minimice el coste total, pero para evitar congestión en la red los paquetes deben mantenerse siempre en el camino directo entre su origen y su destino. Tu objetivo es encontrar los costes mínimos para Q paquetes que se enviarán por esta red.

Implementación

Deberás presentar un único fichero `.cpp`.

📁 Entre los archivos adjuntos encontrarás una plantilla `multihop.cpp` con una implementación de ejemplo.

Tienes que implementar las siguiente funciones:

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

- El entero N representa el número de estaciones de enrutado.
- El entero C representa el coste incremental por unidad de potencia cuando se envía a alta potencia, como se describe arriba.
- El vector A , indizado de 0 a $N - 1$, contiene el coste de enviar a baja potencia desde cada nodo.
- El vector B , indizado de 0 a $N - 1$, contiene el coste base de enviar a alta potencia desde cada nodo, como se describe arriba.
- Los vectores U y V describen las tuberías de la red: hay una tubería entre la estación de enrutado $U[i]$ y la estación de enrutado $V[i]$.
- `query` debe retornar el coste mínimo de enviar un paquete desde la estación de enrutado X hasta la estación de enrutado Y .

El grader llamará a la función `init`, y después llamará a `query` Q veces, imprimiendo su valor de retorno en el archivo de salida.

Sample Grader

El directorio de la tarea contiene una versión simplificada del grader del jurado, que puedes utilizar para probar tu solución localmente. El grader simplificado lee los datos de entrada de `stdin`, llama a las funciones que debes implementar, y finalmente escribe la salida a `stdout`.

La entrada consiste en $N + Q + 2$ líneas:

- Línea 1: los enteros N, Q, C .
- Línea 2: los enteros A_i , separados por espacios.
- Línea 3: los enteros B_i , separados por espacios.
- Línea $4 + i$ ($0 \leq i < N - 1$): los enteros U_i, V_i .
- Línea $4 + (N - 1) + i$ ($0 \leq i < Q$): los enteros X_i, Y_i .

La salida contiene Q líneas, que contienen los valores retornados por la función `query`.

Restricciones

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

Puntuación

Tu programa será probado en un conjunto de casos de prueba agrupados por subtarea. Para obtener la puntuación asociada a una subtarea, debes resolver correctamente todos los casos de prueba que contiene.

- **Subtask 1** [0 puntos]: Casos de ejemplo.
- **Subtask 2** [5 puntos]: $A_i \leq 10, B_i \leq 10$ para cada $i = 0, \dots, N - 1, C \leq 10, N \leq 10, Q \leq 10$.
- **Subtask 3** [10 puntos]: $N \leq 5000, Q = 1$.
- **Subtask 4** [25 puntos]: $N \leq 100\,000, Q = 1$.
- **Subtask 5** [25 puntos]: $N \leq 5000$.
- **Subtask 6** [35 puntos]: Sin restricciones adicionales.

Ejemplos de entrada/salida

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

Explicación

En el **primer caso de ejemplo**, podemos enviar el paquete desde la estación 0 a la 4 a alta potencia, con un coste de 14, y después de 4 a 1 a baja potencia, con un coste de 2. El coste final es 16, que es óptimo.