

JOI '14 Open P1 - Factories

Time limit: 4.0s **Memory limit:** 512M

Contest Day 1 - JOI Open Contest

In IOI Kingdom, there are N cities numbered from 0 to $N - 1$. These cities are connected by $N - 1$ roads through which you can pass in both directions. You can travel from any city to any other city by passing through some of these roads.

In IOI Kingdom, there are many companies producing special components. Each company produces only one kind of components. No two companies produce the same kind of components. Each company has at least one factory. Each factory is built in one of the cities. More than one company may have factories in the same city.

Sometimes, a company requires components of another company. Assume the company C_A requires the components of the company C_B ($C_A \neq C_B$). In this case, they need to transport components from C_B to C_A . They may transport components from any of the factories of the company C_B to any of the factories of the company C_A . They need to choose factories appropriately to minimize the distance between factories.

Task

First, the number of cities and the information of roads in IOI Kingdom are given. Then, Q queries are given. Each query is written in the following form: the company U_j having factories in cities $X_{j,0}, \dots, X_{j,S_j-1}$ requires components of the company V_j having factories in cities $Y_{j,0}, \dots, Y_{j,T_j-1}$. Write a program which, for each query, returns the minimum distance to transport the components.

Implementation Details

You are to write a program which implements procedures to answer queries.

Your program should include the header file `factories.h` by `#include "factories.h"`

Your program should implement the following procedures.

```
void Init(int N, int A[], int B[], int D[])
```

- This procedure is called only once in the beginning. The parameter N is the number of cities in IOI Kingdom. The parameters A , B and D are arrays of length $N - 1$. The elements $A[i]$, $B[i]$ and $D[i]$ are three integers A_i , B_i and D_i ($0 \leq i \leq N - 2$) respectively. This means, for each i ($0 \leq i \leq N - 2$), there is a road of length D_i connecting the city A_i and the city B_i .

```
long long Query(int S, int X[], int T, int Y[])
```

- This procedure is called for each of Q queries. In the query j , the parameters S and T are two integers S_j and T_j respectively. These are the numbers of cities where the companies U_j , V_j have factories respectively. The parameter

X is an array of length S_j . The company U_j has factories in cities $X[0], X[1], \dots, X[S - 1]$. The parameter Y is an array of length T_j . The company V_j has factories in cities $Y[0], Y[1], \dots, Y[T - 1]$.

This procedure should return the minimum distance to transport components from the company V_j to the company U_j .

Constraints

All input data satisfy the following conditions:

- $2 \leq N \leq 500\,000$.
- $1 \leq Q \leq 100\,000$.
- $0 \leq A_i \leq N - 1$ ($0 \leq i \leq N - 2$).
- $0 \leq B_i \leq N - 1$ ($0 \leq i \leq N - 2$).
- $1 \leq D_i \leq 100\,000\,000$ ($0 \leq i \leq N - 2$).
- $A_i \neq B_i$ ($1 \leq i \leq N - 2$).
- You can travel from any city to any other city through some of these roads.
- $1 \leq S_j \leq N - 1$ ($0 \leq j \leq Q - 1$).
- $0 \leq X_{j,k} \leq N - 1$ ($0 \leq j \leq Q - 1, 0 \leq k \leq S_j - 1$).
- $1 \leq T_j \leq N - 1$ ($0 \leq j \leq Q - 1$).
- $0 \leq Y_{j,k} \leq N - 1$ ($0 \leq j \leq Q - 1, 0 \leq k \leq T_j - 1$).
- $X_{j,0}, X_{j,1}, \dots, X_{j,S_j-1}, Y_{j,0}, Y_{j,1}, \dots, Y_{j,T_j-1}$ are different from each other ($0 \leq j \leq Q - 1$).
- $S_0 + S_1 + \dots + S_{Q-1} \leq 1\,000\,000$.
- $T_0 + T_1 + \dots + T_{Q-1} \leq 1\,000\,000$.

Subtask 1 [15 points]

The following conditions are satisfied:

- $N \leq 5\,000$.
- $Q \leq 5\,000$.

Subtask 2 [18 points]

The following conditions are satisfied.

- $S_i \leq 20$ ($0 \leq i \leq Q - 1$).
- $T_i \leq 20$ ($0 \leq i \leq Q - 1$).

Subtask 3 [67 points]

No additional constraints.

Sample Execution

```
Init(7, {0, 1, 2, 2, 4, 1}, {1, 2, 3, 4, 5, 6}, {4, 4, 5, 6, 5, 3});
Query(2, {0, 6}, 2, {3, 4}); //returns 12.
Query(3, {0, 1, 3}, 2, {4, 6}); //returns 3.
Query(1, {2}, 1, {5}); //returns 11.
```

Explanation for Sample

These are sample input and sample output of the sample grader.

- In query 0, the company U_0 has factories in the cities 0, 6, and the company V_0 has factories in the cities 3, 4. The distance from the factory of the company V_0 in the city 3 to the factory of the company U_0 in the city 6 is minimum. The minimum distance is 12.
- In the query 1, the company U_1 has factories in the cities 0, 1, 3, and the company V_1 has factories in the cities 4, 6. The distance from the factory of the company V_1 in the city 6 to the factory of the company U_1 in the city 1 is minimum. The minimum distance is 3.
- In the query 2, the company U_2 has factories in the city 2, and the company V_2 has factories in the city 5. The distance from the factory of the company V_2 in the city 5 to the factory of the company U_2 in the city 2 is minimum. The minimum distance is 11.

Additional Sources

Since the official page does not seem to provide a grader, the following are made available for you to use.

[factories.h](#)

[factories.cpp](#)

[grader.cpp](#)

The code in `factories.h` includes the prototype functions that you need to implement, but do not write your code in this file. The code in `factories.cpp` includes a sample of the functions you should fill in to solve the problem. You should submit the contents of this file to the judge for grading. The code in `grader.cpp` includes the main function and should be used to run judge your program.

If you are using `g++`, then you can compile the program with `g++ factories.cpp grader.cpp -std=c++11`. If you are using an IDE, then consult Stack Overflow on how to add files into your project.