



## Factories

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 the 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  $U_j$ .



## Compilation and Test Run

You can download an archive file from the contest webpage which contains a sample grader to test your program. The archive file also contains a sample source file of your program.

A sample grader consists of one source file which is either `grader.c` or `grader.cpp`. For example, if your program is `factories.c` or `factories.cpp`, you run the following commands to compile your program.

- C

```
gcc -O2 -std=c11 -o grader grader.c factories.c -lm
```

- C++

```
g++ -O2 -std=c++11 -o grader grader.cpp factories.cpp
```

When the compilation succeeds, the executable file `grader` is generated.

Note that the actual grader is different from the sample grader. The sample grader will be executed as a single process, which will read input data from the standard input and write the results to the standard output.

## Input for the sample grader

The sample grader reads the following data from the standard input.

- The first line contains two space separated integers  $N, Q$ , which means there are  $N$  cities in IOI Kingdom, and  $Q$  queries are given to your program.
- The  $(i + 1)$ -st line ( $0 \leq i \leq N - 2$ ) of the following  $(N - 1)$  lines contains three space separated integers  $A_i, B_i, D_i$ , which means there is a road of length  $D_i$  connecting the city  $A_i$  and the city  $B_i$ .
- The information of the  $j$ -th query is written from the  $(3j + 1)$ -st line to the  $(3j + 3)$ -rd line ( $0 \leq j \leq Q - 1$ ) of the following  $3Q$  lines.

The  $(3j + 1)$ -st line ( $0 \leq j \leq Q - 1$ ) contains two space separated integers  $S_j$  and  $T_j$  ( $1 \leq S_j \leq N - 1, 1 \leq T_j \leq N - 1$ ), which means the companies  $U_j$  and  $V_j$  have factories in  $S_j$  and  $T_j$  cities respectively.

The  $(3j + 2)$ -nd line ( $0 \leq j \leq Q - 1$ ) contains  $S_j$  space separated integers  $X_{j,0}, \dots, X_{j,S_j-1}$ , which means the company  $U_j$  has factories in the cities  $X_{j,0}, \dots, X_{j,S_j-1}$ .

The  $(3j + 3)$ -rd line ( $0 \leq j \leq Q - 1$ ) contains  $T_j$  space separated integers  $Y_{j,0}, \dots, Y_{j,T_j-1}$ , which means the company  $V_j$  has factories in the cities  $Y_{j,0}, \dots, Y_{j,T_j-1}$ .

## Output of the sample grader

When the program terminates successfully, the sample grader writes to the **standard output** the values returned by Query one per line.



## 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

### 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.

- $S_i \leq 10$  ( $0 \leq i \leq Q - 1$ ).
- $T_i \leq 10$  ( $0 \leq i \leq Q - 1$ ).

**Subtask 3 [67 points]**

There are no additional constraints.

**Sample Input and Output**

Sample Input	Sample Output
7 3	12
0 1 4	3
1 2 4	11
2 3 5	
2 4 6	
4 5 5	
1 6 3	
2 2	
0 6	
3 4	
3 2	
0 1 3	
4 6	
1 1	
2	
5	

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

- In the 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.