Spy 3

Aoi and Bitaro are spies belonging to the National Information Bureau of JOI country. Their mission this time is to conduct undercover investigation into the IOI country. Bitaro infiltrates into the IOI country, while Aoi gives instructions to Bitaro from the JOI country.

Before the infiltration, Aoi and Bitaro obtained a map of the IOI country. The IOI country has N cities, numbered from 0 to $N-1$. Moreover, there are M roads in the IOI country, numbered from 0 to $M-1$. Road i ($0 \leq i \leq M-1$) connects city A_i and city B_i bidirectionally, with a length of C_i. It is possible to travel between any pair of cities by passing some roads. Bitaro moves between cities in the IOI country by passing these roads.

Additionally, there are Q investigation plans. The j-th plan ($0 \leq j \leq Q-1$) involves investigating city T_j in the IOI country.

All the information above were first given to both Aoi and Bitaro. Then, Bitaro proceeded with the infiltration into the IOI country.

Bitaro successfully evaded numerous pursuers, defeated assassins, and finally managed to infiltrate into city 0 of the IOI country. However, due to the extremely difficult nature of the infiltration operation, Bitaro lost some of the information about the IOI country. Specifically, Bitaro lost information about the length of K roads, namely the roads $X_0, X_1, \ldots, X_{K-1}$. In other words, Bitaro no longer knows the values of $C_{X_0}, C_{X_1}, \ldots, C_{X_{K-1}}$. Note that while Bitaro lost this information, Aoi still retains it.

Bitaro immediately reported to Aoi which road lengths' information he had lost.

To accomplish the mission, Bitaro wants to know a shortest path from city 0 to each of the cities targeted by the Q investigation plans.

Aoi will send Bitaro a string where each character is either '`0`' or '`1`' to assist him. Due to the risk of interception, Aoi wants to minimize the content sent to Bitaro.

Write a program to implement Aoi's strategy for sending a string to Bitaro, given the information about the IOI country, the investigation plans, and the roads Bitaro lost information about. Also, write a program to implement Bitaro's strategy for finding the shortest paths given the information he possesses and the string sent by Aoi.

Implementation Details

You should submit 2 files.

The first file is `Aoi.cpp`. This file is intended to implement Aoi's strategy and should implement the following functions. The program should include `Aoi.h` using the preprocessor directive `#include`.

- ```
  std::string aoi(int N, int M, int Q, int K, std::vector<int> A,
                  std::vector<int> B, std::vector<long long> C,
                  std::vector<int> T, std::vector<int> X)
  ```

  This function is called only once for each test case.
  - The parameter `N` is the number of cities in IOI Kingdom N.
  - The parameter `M` is the number of roads in IOI Kingdom M.
  - The parameter `Q` is the number of investigation plans Q.
  - The parameter `K` is the number of roads that Bitaro lost the length information of, denoted by K.
  - The parameters `A`, `B`, `C` are arrays of length M. They mean the road i $(0 \leq i \leq M-1)$ connects the city `A[i]` and the city `B[i]` bidirectionally, with a length of `C[i]`.
  - The parameter `T` is an array of length Q. It means the j-th plan $(0 \leq j \leq Q-1)$ involves investigating city `T[j]`.
  - The parameter `X` is an array of length K. It indicates that Bitaro lost the length information of roads `X[0]`, `X[1]`, ..., `X[K-1]`.
  - The return value is the string s that Aoi sends to Bitaro.
  - Each character of the string s must be either '`0`' or '`1`'. If this condition is not met, your program will be judged as **Wrong Answer [1]**.
  - The length of string s must be at most 12 000. If this condition is not met, your program will be judged as **Wrong Answer [2]**.

The second file is `Bitaro.cpp`. This file is intended to implement Bitaro's strategy and should implement the following functions. The program should include `Bitaro.h` using the preprocessor directive `#include`.
  ```

- `void bitaro(int N, int M, int Q, int K, std::vector<int> A, std::vector<int> B, std::vector<long long> C, std::vector<int> T, std::vector<int> X, std::string s)`

  This function is called only once after the function `aoi` is called.
  - The parameter `N` is the number of cities in IOI Kingdom N.
  - The parameter `M` is the number of roads in IOI Kingdom M.
  - The parameter `Q` is the number of investigation plans Q.
  - The parameter `K` is the number of roads that Bitaro lost the length information of, denoted by K.
  - The parameters `A`, `B`, `C` are arrays of length M. They mean the road i ($0 \leq i \leq M-1$) connects the city `A[i]` and the city `B[i]` bidirectionally, with a length of `C[i]`.
  - The parameter `T` is an array of length Q. It means the j-th plan ($0 \leq j \leq Q-1$) involves investigating city `T[j]`.
  - The parameter `X` is an array of length K. It indicates that Bitaro lost the length information of roads `X[0]`, `X[1]`, ..., `X[K-1]`.
  - The argument `s` is a string where each character is either '`0`' or '`1`', representing the string sent by Aoi to Bitaro.

Within `Bitaro.cpp`, your program can call the following function.

★ `void answer(const std::vector<int> &e)`

In the $(j+1)$-th call ($0 \leq j \leq Q-1$) to this function, you have to answer the shortest path from city 0 to city T_j, which is the target of the j-th investigation plan.

- ◇ The parameter `e` is an array that represents the shortest path from city 0 to city T_j.
- ◇ Let n be the length of the array `e`. The elements `e[0]`, `e[1]`, ..., `e[n-1]` are the indices of roads in the shortest path from city 0 to city T_j, in the order of traversal.
- ◇ If there are multiple possible shortest paths, any of them can be chosen as the answer.
- ◇ Each element of the parameter `e` must be between 0 and $M-1$. If this condition is not met, your program will be judged as **Wrong Answer [3]**.
- ◇ The sequence of roads indicated by the argument `e` must form a path from city 0 to city T_j. More formally, it must satisfy the following conditions.

  There exists a sequence of numbers $u_0, u_1, \ldots, u_n$ such that
  - $u_0 = 0$.
  - $u_n = T_j$.
  - The road `e[k]` (where $0 \leq k \leq n-1$) connects city u_k and city u_{k+1}.

  If these conditions are not met, your program will be judged as **Wrong Answer [4]**.

- ◇ If the sequence of roads indicated by the argument `e` is not the shortest path from city 0 to city T_j among all paths starting from city 0 and ending at city T_j, your program will be judged as **Wrong Answer [5]**. Here, the length of the path is defined as the sum of the lengths of the roads used.
- ◇ The function `answer` must be called exactly Q times. If the number of calls to the function `answer` is not equal to Q at the end of the execution of the function `bitaro`, your program will be judged as **Wrong Answer [6]**.

Important Notices

- Your program can implement other functions for internal use, or use global variables. Submitted files will be compiled with the grader, and become a single executable file. All global variables and internal functions should be declared in an unnamed namespace to avoid confliction with other files. When it is graded, it will be executed as two processes of Aoi and Bitaro. The process of Aoi and the process of Bitaro cannot share global variables.
- Your program must not use the standard input and the standard output. Your program must not communicate with other files by any methods. However, your program may output debugging information to the standard error.

Compilation and Test Run

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

The sample grader is the file `grader.cpp`. In order to test your program, put `grader.cpp`, `Aoi.cpp`, `Bitaro.cpp`, `Aoi.h`, `Bitaro.h` in the same directory, and run the following command to compile your programs. Instead, you may run `compile.sh` contained in the archive file.

```
g++ -std=gnu++20 -O2 -o grader grader.cpp Aoi.cpp Bitaro.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 and the standard error output.

Input for the Sample Grader

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

$N\ M$
$A_0\ B_0\ C_0$
$A_1\ B_1\ C_1$
$\vdots$
$A_{M-1}\ B_{M-1}\ C_{M-1}$
Q
$T_0\ T_1\ \ldots\ T_{Q-1}$
K
$X_0\ X_1\ \ldots\ X_{K-1}$

Output of the Sample Grader

The sample grader outputs the following information to the standard output and the standard error output (quotes for clarity).

- If your program is judged as **Wrong Answer [1], [2], [3], [4], or [6]**, the sample grader writes its type as "`Wrong Answer [1]`" in the standard error output. Nothing will be output to the standard output.
- If not, the length of the returned string s from the function `aoi` will be output to the standard error output in the format "`Accepted: 2024`". Additionally, the length of the path in the $(j+1)$-th call $(0 \leq j \leq Q-1)$ to the `answer` function will be output to $(j+1)$-th line of the standard output. The sample grading program **does not check** whether the path is the shortest.

If your program meets the conditions of several types of Wrong Answer, the sample grader reports only one of them.

Constraints

All input data satisfies the following conditions:

- $2 \leq N \leq 10\,000$.
- $1 \leq M \leq 20\,000$.
- $1 \leq Q \leq 16$.

- $1 \leq K \leq 300$.
- $0 \leq A_i < B_i \leq N - 1$ $(0 \leq i \leq M - 1)$.
- $(A_i, B_i) \neq (A_j, B_j)$ $(0 \leq i < j \leq M - 1)$.
- $1 \leq C_i \leq 10^{12}$ $(0 \leq i \leq M - 1)$.
- $1 \leq T_j \leq N - 1$ $(0 \leq j \leq Q - 1)$.
- $T_j \neq T_k$ $(0 \leq j < k \leq Q - 1)$.
- $0 \leq X_k \leq M - 1$ $(0 \leq k \leq K - 1)$.
- $X_k \neq X_l$ $(0 \leq k < l \leq K - 1)$.
- You can travel from any city to any other city by traversing some roads.
- All input values are integers.

Grading

If your program is judged as any type of **Wrong Answer [1] - [6]** (see Implementation Details) or any type of runtime errors (TLE (Time Limit Exceeded), MLE (Memory Limit Exceeded), Abnormal End, etc.) in any of the test cases, your score is 0 points.

Otherwise, the grading is based on the maximum length of the returned string s from the function `aoi`, denoted as L, across all test cases for this task.

- If $1561 \leq L \leq 12\,000$, the score is $\left\lfloor \frac{100\,000}{L - 560} \right\rfloor$ points.
- If $0 \leq L \leq 1560$, the score is 100 points.

Here, $\lfloor x \rfloor$ represents the largest integer that is not greater than x.

Sample Communication

Here is a sample input for the sample grader and corresponding function calls.

<table>
<tr><td rowspan="2">Sample Input 1</td><td colspan="3">Sample Function Calls</td></tr>
<tr><td>Call</td><td>Call</td><td>Return value</td></tr>
<tr><td>3 3
1 2 1
0 2 2
0 1 3
2
2 1
2
0 1</td><td>aoi(3, 3, 2, 2, [1, 0, 0], [2, 2, 1],
[1, 2, 3], [2, 1], [0, 1])</td><td></td><td>

"101001"</td></tr>
<tr><td>bitaro(3, 3, 2, 2, [1, 0, 0], [2, 2, 1],
[-1, -1, 3], [2, 1], [0, 1], "101001")</td><td></td><td></td></tr>
<tr><td></td><td>answer([1])</td><td></td></tr>
<tr><td></td><td>answer([1, 0])</td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

The shortest path from city 0 to city 1, is achieved either by traversing roads 1 and 0 in this order, or simply passing through road 2. Therefore, for the second call to the `answer` function in this sample, it is also acceptable to call `answer([2])`.

The file that can be downloaded from the contest site, `sample-01-in.txt`, corresponds to input example 1.

The files `sample-01-in.txt` and `sample-02-in.txt` included in the downloadable files from the contest site can be used as input for the sample grader.