



## スパイ 3 (Spy 3)

葵とビ太郎は JOI 国情報管理局に所属するスパイである。

今回のミッションは IOI 国への潜入調査である。ビ太郎は IOI 国に潜入し、葵は JOI 国からビ太郎に指示を出す。

潜入の前に葵とビ太郎は IOI 国の地図を入手した。IOI 国には  $N$  個の都市があり、 $0$  から  $N-1$  までの番号が付けられている。また、IOI 国には  $M$  本の道路があり、 $0$  から  $M-1$  までの番号が付けられている。道路  $i$  ( $0 \leq i \leq M-1$ ) は都市  $A_i$  と都市  $B_i$  を双方向に結んでおり、その長さは  $C_i$  である。どの都市からどの都市へも何本かの道路を通ることで移動できる。ビ太郎はこの道路を通ることで IOI 国の都市間を移動する。

また、 $Q$  個の調査計画がある。 $j$  番目 ( $0 \leq j \leq Q-1$ ) の計画は、IOI 国の都市  $T_j$  を調査するというものである。

葵、ビ太郎ともに以上の情報を持った上で、ビ太郎は IOI 国への潜入に臨んだ。

ビ太郎は数々の追っ手を撒き、刺客を返り討ちにし、ついに IOI 国の都市  $0$  に潜入することに成功した。しかしながら、潜入作戦はとても困難であったため、ビ太郎は IOI 国の情報の一部を失ってしまった。具体的には、 $K$  本の道路、道路  $X_0, X_1, \dots, X_{K-1}$  の長さの情報を失ってしまった。すなわち、ビ太郎は  $C_{X_0}, C_{X_1}, \dots, C_{X_{K-1}}$  の値がわからなくなってしまった。情報を失ったのはビ太郎であり、葵はこれらの情報を保持していることに注意せよ。

ビ太郎はすぐさまどの道路の長さの情報を失ったか葵に報告した。

ビ太郎はミッションを遂行するために、 $Q$  個の調査計画の対象である都市すべてについて、都市  $0$  からの最短経路のうち  $1$  つを知りたい。

葵はビ太郎を助けるために各文字が '0' または '1' である文字列をビ太郎に送信する。通信を傍受される恐れがあるため、ビ太郎に送信する内容は少なくしたい。

IOI 国の情報、調査計画の情報とビ太郎が紛失した道路の情報が与えられたとき、ビ太郎に文字列を送信する葵の戦略を実装したプログラムおよび、ビ太郎が保持している情報と葵から送信された文字列の情報が与えられたとき、最短経路を求めるビ太郎の戦略を実装したプログラムを作成せよ。



## 実装の詳細

あなたは2つのファイルを提出しなければならない。

1つ目のファイルは `Aoi.cpp` という名前である。このファイルは葵の戦略を実装したファイルであり、以下の関数を実装していなければならない。また、`#include` プリプロセッサ指令によって `Aoi.h` を読み込むこと。

- `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)`

この関数は各テストケースにおいて1回だけ呼び出される。

- 引数 `N` は、IOI 国の都市の数  $N$  である。
- 引数 `M` は、IOI 国の道路の本数  $M$  である。
- 引数 `Q` は、調査計画の数  $Q$  である。
- 引数 `K` は、ビ太郎が長さの情報を失ってしまった道路の本数  $K$  である。
- 引数 `A`, `B`, `C` は長さ  $M$  の配列であり、道路  $i$  ( $0 \leq i \leq M-1$ ) が都市 `A[i]`, 都市 `B[i]` を双方向に結んでおり、その長さが `C[i]` であることを表す。
- 引数 `T` は長さ  $Q$  の配列であり、 $j$  番目 ( $0 \leq j \leq Q-1$ ) の計画は都市 `T[j]` を調査するというものであることを表す。
- 引数 `X` は長さ  $K$  の配列であり、ビ太郎が道路 `X[0], X[1], ..., X[K-1]` の長さの情報を失ったことを表す。
- 戻り値は、葵がビ太郎に送信する文字列  $s$  である。
- 文字列  $s$  の各文字は `'0'` または `'1'` でなければならない。これが満たされない場合、不正解 [1] と判定される。
- 文字列  $s$  の長さは 12000 以下でなければならない。これが満たされない場合、不正解 [2] と判定される。

2つ目のファイルは `Bitaro.cpp` という名前である。このファイルはビ太郎の戦略を実装したファイルであり、以下の関数を実装していなければならない。また、`#include` プリプロセッサ指令によって `Bitaro.h` を読み込むこと。



- `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)`

この関数は、関数 `aoi` が呼び出された後に 1 回だけ呼び出される。

- 引数 `N` は、IOI 国の都市の数  $N$  である。
- 引数 `M` は、IOI 国の道路の本数  $M$  である。
- 引数 `Q` は、調査計画の数  $Q$  である。
- 引数 `K` は、ビ太郎が長さの情報を失ってしまった道路の本数  $K$  である。
- 引数 `A`, `B`, `C` は長さ  $M$  の配列であり、道路  $i$  ( $0 \leq i \leq M-1$ ) が都市 `A[i]`, 都市 `B[i]` を双方向に結んでおり、その長さが `C[i]` であることを表す。ただし、ビ太郎が失ってしまった長さの情報、`C[X[0]]`, `C[X[1]]`, ..., `C[X[K-1]]` には長さの値の代わりに `-1` が代入されている。
- 引数 `T` は長さ  $Q$  の配列であり、 $j$  番目 ( $0 \leq j \leq Q-1$ ) の計画は都市 `T[j]` を調査するというものであることを表す。
- 引数 `X` は長さ  $K$  の配列であり、ビ太郎が道路 `X[0]`, `X[1]`, ..., `X[K-1]` の長さの情報を失ったことを表す。
- 引数 `s` は各文字が '0' または '1' である文字列であり、葵がビ太郎に送信した文字列を表す。

`Bitaro.cpp` 内では以下の関数を呼び出すことができる。

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

この関数を呼び出すのが  $j+1$  回目 ( $0 \leq j \leq Q-1$ ) であるとする。この関数を用いて、都市 0 から  $j$  番目の調査計画の対象である都市  $T_j$  への最短経路を解答する。

- ◇ 引数 `e` は、都市 0 から都市  $T_j$  への最短経路を示す配列である。
- ◇ 引数 `e` の長さを  $n$  とするとき、`e[0]`, `e[1]`, ..., `e[n-1]` は、都市 0 から都市  $T_j$  への最短経路において通る道路の番号を、通る順に並べた配列である。
- ◇ 最短経路が複数考えられる場合は、そのうちのどの経路を解答しても良い。
- ◇ 引数 `e` の各要素は、0 以上  $M-1$  以下でなければならない。これが満たされていない場合、不正解 [3] と判定される。
- ◇ 引数 `e` で示された道路の列は、都市 0 から出発し、都市  $T_j$  へ至る経路でなければならない。より形式的には、以下を満たさなければならない。

ある数列  $u_0, u_1, \dots, u_n$  が存在して、

- \*  $u_0 = 0$ .
- \*  $u_n = T_j$ .
- \* 道路 `e[k]` ( $0 \leq k \leq n-1$ ) は都市  $u_k$  と都市  $u_{k+1}$  を結ぶ。

これが満たされていない場合、不正解 [4] と判定される。



- ◇ 引数  $e$  で示された道路の列が、都市  $0$  から出発し、都市  $T_j$  へ至る経路のうち、最短でない場合、不正解 [5] と判定される。ただし、経路の長さは、用いた道路の長さの総和として定義される。
- ◇ 関数 `answer` はちょうど  $Q$  回呼び出される必要がある。関数 `bitaro` の実行の終了時に関数 `answer` の呼び出し回数が  $Q$  回でなかった場合、不正解 [6] と判定される。

## 重要な注意

- 内部での使用のために他の関数を実装したり、グローバル変数を宣言するのは自由である。ただし、提出された2つのプログラムは、採点プログラムとまとめてリンクされて1つの実行ファイルになるので、各ファイル内のすべてのグローバル変数と内部関数を無名名前空間内で宣言して、他のファイルとの干渉を避ける必要がある。採点時には、このプログラムは葵側、ビ太郎側として2個のプロセスとして実行されるので、葵側とビ太郎側でプログラム中のグローバル変数を共有することはできない。
- あなたの提出したプログラムは、標準入力・標準出力、あるいは他のファイルといかなる方法でもやりとりしてはならない。ただし、標準エラー出力にデバッグ情報等を出力することは許される。

## コンパイル・実行の方法

作成したプログラムをテストするための、採点プログラムのサンプルが、コンテストサイトからダウンロードできるアーカイブの中に含まれている。このアーカイブには、提出しなければならないファイルのサンプルも含まれている。

採点プログラムのサンプルは1つのファイルからなる。そのファイルは `grader.cpp` である。作成したプログラムをテストするには、`grader.cpp`, `Aoi.cpp`, `Bitaro.cpp`, `Aoi.h`, `Bitaro.h` を同じディレクトリに置き、次のようにコマンドを実行する。なお、アーカイブの中に含まれている `compile.sh` というファイルを代わりに実行してもよい。

```
g++ -std=gnu++20 -O2 -o grader grader.cpp Aoi.cpp Bitaro.cpp
```

コンパイルが成功すれば、`grader` という実行ファイルが生成される。

実際の採点プログラムは、採点プログラムのサンプルとは異なることに注意せよ。採点プログラムのサンプルは単一のプロセスとして起動する。このプログラムは、標準入力から入力を読み込み、標準出力と標準エラー出力に結果を出力する。



## 採点プログラムのサンプルの入力

採点プログラムのサンプルは標準入力から以下の形式で入力を読み込む。

```
N M
A0 B0 C0
A1 B1 C1
⋮
AM-1 BM-1 CM-1
Q
T0 T1 … TQ-1
K
X0 X1 … XK-1
```

## 採点プログラムのサンプルの出力

採点プログラムのサンプルは標準出力と標準エラー出力へ以下の情報を出力する（引用符は実際には出力されない）。

- 不正解 [1], 不正解 [2], 不正解 [3], 不正解 [4], 不正解 [6] と判定された場合、標準エラー出力に、不正解の種類が “Wrong Answer [1]” のように出力される。標準出力には何も出力されない。
- そうでない場合、標準エラー出力に、関数 `aoi` の戻り値 `s` の長さが “Accepted: 2024” のように出力される。また、標準出力の  $j+1$  行目 ( $0 \leq j \leq Q-1$ ) に  $j+1$  回目の `answer` 関数の呼び出しにおける経路の長さが出力される。採点プログラムのサンプルは、経路が最短であるかは判定しない。

実行するプログラムが複数の不正解の条件を満たした場合、表示される不正解の種類はそれらのうち 1 つのみである。

## 制約

すべての入力データは以下の条件を満たす。

- $2 \leq N \leq 10\,000$ .
- $1 \leq M \leq 20\,000$ .
- $1 \leq Q \leq 16$ .
- $1 \leq K \leq 300$ .



- $0 \leq A_i < B_i \leq N - 1$  ( $0 \leq i \leq M - 1$ ).
- $(A_i, B_i) \neq (A_j, B_j)$  ( $0 \leq i < j \leq M - 1$ ).
- $1 \leq C_i \leq 10^{12}$  ( $0 \leq i \leq M - 1$ ).
- $1 \leq T_j \leq N - 1$  ( $0 \leq j \leq Q - 1$ ).
- $T_j \neq T_k$  ( $0 \leq j < k \leq Q - 1$ ).
- $0 \leq X_k \leq M - 1$  ( $0 \leq k \leq K - 1$ ).
- $X_k \neq X_l$  ( $0 \leq k < l \leq K - 1$ ).
- どの都市からどの都市へも何本かの道路を通ることで移動できる。
- 入力される値はすべて整数である。

## 採点基準

この課題のテストケースの中で、1つでも不正解 [1] ~ [6] (実装の詳細を参照) と判定されたものや、実行時間制限超過、メモリ制限超過、実行時エラーと判定されたものがあつた場合、0点となる。

また、この課題におけるすべてのテストケースに正解した場合、この課題のすべてのテストケースで呼び出された関数 `aoi` の戻り値 `s` の長さの最大値を  $L$  として、以下のように採点される。

- $1561 \leq L \leq 12000$  のとき、 $\left\lfloor \frac{100000}{L - 560} \right\rfloor$  点。
- $0 \leq L \leq 1560$  のとき、100 点。

ここで、 $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。



## やり取りの例

採点プログラムのサンプルが読み込む入力の例と、それに対応する関数の呼び出しの例を以下に示す。

入力例 1	関数の呼び出しの例		
	呼び出し	呼び出し	戻り値
3 3 1 2 1	aoi(3, 3, 2, 2, [1, 0, 0], [2, 2, 1], [1, 2, 3], [2, 1], [0, 1])		"101001"
0 2 2 0 1 3	bitaro(3, 3, 2, 2, [1, 0, 0], [2, 2, 1], [-1, -1, 3], [2, 1], [0, 1], "101001")		
2		answer([1])	
2 1		answer([1, 0])	
2 0 1			

都市 0 から都市 1 への最短経路は、道路 1, 道路 0 をこの順に通る経路と、道路 2 を通る経路の 2 通りが考えられる。そのため、この入力例における 2 回目の `answer` 関数の呼び出しは、`answer([2])` と呼び出してよい。

コンテストサイトからダウンロードできるファイルのうち、`sample-01-in.txt` は入力例 1 に対応する。

コンテストサイトからダウンロードできるファイルに含まれる `sample-01-in.txt`, `sample-02-in.txt` は、採点プログラムのサンプルの入力として用いることができる。