



ビーバーの会合 (Meetings)

ビーバーが住んでいる島が N 個あり、 0 から $N-1$ の番号が付けられている。これらの島は $N-1$ 本の双方向に通行可能な橋で結ばれており、どの島の間もいくつかの橋を通して移動が可能である。どの島に対しても、その島に直接つながっている橋の本数は 18 本以下である。それぞれの島には 1 匹のビーバーが住んでいる。

時々、いくつかの島に住むビーバーが、 1 つの島に集まって会合を行う。集まるビーバーがちょうど 3 匹のときは、ビーバーが集まる島は次のような島である：

集まる際に、 3 匹のビーバーが自分の住む島から移動する際に通る橋の本数の合計が最小になるような島 (このような島はただ一つ存在する)。

この島は、 3 匹のいずれかのビーバーが住む島であるかもしれないことに注意せよ。

あなたは、 N 個の島がどのように橋で結ばれているかが気になっている。あなたは直接島に行って橋の様子を確認することができないため、ビーバーに次のような指示を出して、島のつながり方を知ろうと考えた：

- 3 つの島 u, v, w ($0 \leq u \leq N-1, 0 \leq v \leq N-1, 0 \leq w \leq N-1, u \neq v, u \neq w, v \neq w$) を指定し、島 u 、島 v 、島 w に住むビーバーに会合を行わせる。
- このとき、あなたは 3 匹のビーバーが集まった島の番号を知ることができる。

あなたは、ビーバーたちに少ない回数の指示を送ることで、島のつながり方を特定したい。

島の数が与えられたとき、ビーバーたちとやりとりを行うことで、島のつながり方を特定するプログラムを作成せよ。

実装の詳細

あなたは 1 つのファイルを提出しなければならない。

あなたの提出するファイルは `meetings.cpp` という名前である。このファイルは以下の関数を実装していなければならない。そのプログラムは `meetings.h` をインクルードすること。

- `void Solve(int N)`

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 は島の数 N を表す。

あなたのプログラムは以下の関数を呼び出すことができる。

- ★ `int Query(int u, int v, int w)`



この関数は、3つの島の番号を指定すると、それらの島に住むビーバーたちで会合を行うときに集まる島の番号を返す。

◇ 3つの島の番号 u, v, w をそれぞれ引数 u, v, w で指定する。これらは、 $0 \leq u \leq N-1, 0 \leq v \leq N-1, 0 \leq w \leq N-1, u \neq v, u \neq w, v \neq w$ を満たさなければならない。満たされない場合、不正解 [1] と判定される。

◇ 関数 `Query` を 100 000 回を超えて呼び出してはならない。100 000 回を超えて呼び出した場合、不正解 [2] と判定される。

★ `void Bridge(int u, int v)`

この関数を用いて、島が橋でどのように結ばれているかを回答する。

◇ 引数 u, v は、島 u と島 v が橋で直接結ばれていることを表す。

◇ $0 \leq u < v \leq N-1$ が成り立たない場合、不正解 [3] と判定される。

◇ 島 u と島 v を直接結ぶ橋が存在しない場合、不正解 [4] と判定される。

◇ 関数 `Bridge` を同じ引数 u, v で複数回呼び出した場合、不正解 [5] と判定される。

◇ 橋は $N-1$ 本存在するため、関数 `Bridge` はちょうど $N-1$ 回呼び出される必要がある。関数 `Solve` の実行の終了時に関数 `Bridge` の呼び出し回数が $N-1$ 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。
- 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、これらのファイル `grader.cpp, meetings.cpp, meetings.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。

```
g++ -std=gnu++14 -O2 -o grader grader.cpp meetings.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すること。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入力を読み込む。

```
N
A0 B0
⋮
AN-2 BN-2
```

A_i, B_i ($0 \leq i \leq N-2$) は、島 A_i と島 B_i が直接橋で結ばれていることを表す。

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)。

- 正解の場合、関数 `Query` の呼び出し回数が “Accepted: 100” のように出力される。
- 不正解の場合、不正解の種類が “Wrong Answer [1]” のように出力される。

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

A_i, B_i の意味については「採点プログラムのサンプルの入力」の項目を参照せよ。

- $3 \leq N \leq 2000$.
- $0 \leq A_i < B_i \leq N-1$ ($0 \leq i \leq N-2$).
- どの島の間も、いくつかの橋を通して移動可能である。
- どの島に対しても、その島に直接つながっている橋の本数は 18 本以下である。



小課題

1. (7 点) $N \leq 7$.
 2. (10 点) $N \leq 50$.
 3. (12 点) $N \leq 300$.
 4. (71 点) 追加の制約はない.
- 小課題 1, 2, 3 では, 各小課題のすべてのテストケースに正解すると, 満点が与えられる.
 - 小課題 4 では, すべてのテストケースに正解したとき, 各テストケースに対する関数 `Query` の呼び出し回数の最大値を X として, 以下のように採点される.
 - $40\,000 < X \leq 100\,000$ のとき, 49 点.
 - $X \leq 40\,000$ のとき, 71 点.

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
5	<code>Solve(5)</code>		
0 1		<code>Query(0, 1, 2)</code>	0
0 2		<code>Query(0, 3, 4)</code>	1
1 3		<code>Bridge(1, 3)</code>	(なし)
1 4		<code>Bridge(0, 2)</code>	(なし)
		<code>Bridge(1, 4)</code>	(なし)
		<code>Bridge(0, 1)</code>	(なし)