



ベルトコンベア (Belt Conveyor)

JOI 社の工場には N 個の台があり、 0 から $N-1$ までの番号が付けられている。また、工場には $N-1$ 本のベルトコンベアがあり、 0 から $N-2$ までの番号が付けられている。ベルトコンベア i ($0 \leq i \leq N-2$) は台 A_i と台 B_i のいずれか一方から他方へ製品を運搬するが、どちらの向きに運搬するか見ることはできない。ベルトコンベアの向きを無視すると、どの 2 つの台も、いくつかのベルトコンベアを通じて繋がっている。

工場長の IOI 君はそれぞれのベルトコンベアがどちらの向きに製品を運搬するか忘れてしまったため、以下の一連の操作を何回か行うことにした。

1. いくつかのベルトコンベアを選び、それらの運搬の向きを反転させる。
2. いくつかの台を選び、それらの台に製品を 1 つずつ置く。
3. 製品が置かれているそれぞれの台について同時に、次のことが起こる。
 - その台から他の台に向かうベルトコンベアがない場合は、何も起こらない。
 - その台から他の台に向かうベルトコンベアがある場合は、製品がそのうちの 1 本に乗って移動し、移動した先の台で停止する。このあと、製品が再び移動することはない。
4. IOI 君はそれぞれの台に製品が 1 つ以上存在するか確認し、製品をすべて回収する。
5. 1. で反転させたベルトコンベアの向きを元に戻す。

IOI 君はこの操作を 30 回以下行うことで、それぞれのベルトコンベアが最初どちらを向いていたかを特定したい。

それぞれのベルトコンベアが繋ぐ台の情報が与えられたとき、30 回以下の操作を行うことでそれぞれのベルトコンベアの向きを特定する IOI 君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 `conveyor.cpp` という名前である。そのプログラムは `#include` プリプロセッサ指令によって `conveyor.h` を読み込むこと。

`conveyor.cpp` は以下の関数を実装していなければならない。

- `void Solve(int N, std::vector<int> A, std::vector<int> B)`

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 は台の個数 N である。
- 引数 A, B は長さ $N-1$ の配列であり、ベルトコンベアが繋ぐ台の情報を表す。

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。

- ★ `std::vector<int> Query(std::vector<int> x, std::vector<int> y)`

この関数を用いて、工場での操作を行う。

- 引数 x は長さ $N-1$ の配列である。 $0 \leq i \leq N-2$ について、 $x[i]$ が1のときベルトコンベア i の向きを反転させることを、0のとき反転させないことを表す。
- 引数 y は長さ N の配列である。 $0 \leq j \leq N-1$ について、 $y[j]$ が1のとき台 j に製品を置くことを、0のとき置かないことを表す。
- 戻り値を z とすると、 z は長さ N の配列である。 $0 \leq j \leq N-1$ について、 $z[j]$ が1のとき台 j に製品が1つ以上存在することを、0のとき存在しないことを表す。
- 引数 x の長さは $N-1$ 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 [1] と判定される。
- 引数 x の各要素は、0または1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 [2] と判定される。
- 引数 y の長さは N 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 y の各要素は、0または1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 [4] と判定される。
- 関数 `Query` を30回を超えて呼び出してはならない。30回を超えて呼び出した場合、不正解 [5] と判定される。

- ★ `void Answer(std::vector<int> a)`

この関数を用いて、ベルトコンベアの最初の向きを解答する。

- 引数 a は長さ $N-1$ の配列である。 $0 \leq i \leq N-2$ について、 $a[i]$ が0のときベルトコンベア i が A_i から B_i に製品を運搬していることを、1のとき B_i から A_i に運搬していることを



表す。

- ◇ 引数 a の長さは $N - 1$ 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 [6] と判定される。
- ◇ 引数 a の各要素は、0 または 1 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 [7] と判定される。
- ◇ 誤った向きを回答した場合、不正解 [8] と判定される。
- ◇ 関数 `Answer` は丁度 1 回呼び出さなければならない。関数 `Answer` を 2 回以上呼び出した場合、不正解 [9] と判定される。関数 `Solve` の実行の終了時に関数 `Answer` が 1 回も呼び出されていなかった場合、不正解 [10] 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`, `conveyor.cpp`, `conveyor.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++17 -O2 -o grader grader.cpp conveyor.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形式で入力を読み込む。

```
N
A0 A1 ⋯ AN-2
B0 B1 ⋯ BN-2
C0 C1 ⋯ CN-2
```

C_i ($0 \leq i \leq N-2$) は、ベルトコンベア i が台 A_i から台 B_i に向けて製品を運搬するとき 0、台 B_i から台 A_i に運搬するとき 1 である。

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: 22” のように出力される。
- 不正解の場合、不正解の種類が “Wrong Answer [4]” 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。

採点プログラムのサンプルにおいては、それぞれの製品の移動に用いられるベルトコンベアは、製品がある台から他の台に向かうベルトコンベアのうち 1 本が、実行ごとに結果が変わらない疑似乱数を用いて一様に無作為に選ばれる。シードの値を変更したい場合は、

```
./grader 2023
```

のように 1 個目の引数に整数値を与えて実行せよ。

採点に関する注意

いくつかのテストケースについて、実際の採点プログラムは適応的 (adaptive) である。これは、採点プログラムは初めから固定された答えを持たず、それ以前の `Query` 関数の呼び出しに応じて採点プログラムが応答するということである。ただし、すべての応答に矛盾しないような答えが少なくとも 1 つ存在することが保証される。



制約

すべての入力データは以下の条件を満たす。

- $0 \leq A_i \leq N - 1$ ($0 \leq i \leq N - 2$).
- $0 \leq B_i \leq N - 1$ ($0 \leq i \leq N - 2$).
- ベルトコンベアの向きを無視すると、どの2つの台も、いくつかのベルトコンベアを通じて繋がっている。



小課題

1. (5 点) $N = 2$
2. (20 点) $N = 30$.
3. (15 点) $N = 100\,000$, $A_i = i$ ($0 \leq i \leq N - 2$), $B_i = i + 1$ ($0 \leq i \leq N - 2$).
4. (60 点) $N = 100\,000$.

やりとりの例

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	Solve(3, [0, 2], [2, 1])		
0 2		Query([0, 0], [0, 0, 1])	[1, 0, 0]
2 1		Query([1, 0], [1, 0, 1])	[0, 1, 1]
1 0		Query([1, 1], [0, 0, 1])	[0, 0, 1]
		Query([0, 1], [1, 1, 1])	[1, 0, 1]
		Answer([1, 0])	

1 回目の Query の呼び出しの戻り値は、[1, 0, 0] の他に [0, 1, 0] となる可能性もある。

2 回目の Query の呼び出しにおいて、台 0 に置かれた製品はベルトコンベア 0 を通って台 2 に移動して停止する。この製品が続けてベルトコンベア 1 を通って台 1 に移動することはないことに注意すること。

この入力例はどの小課題の制約も満たさないことに注意すること。

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-02.txt は小課題 1 の制約を満たし、sample-03.txt は小課題 2 の制約を満たす。