



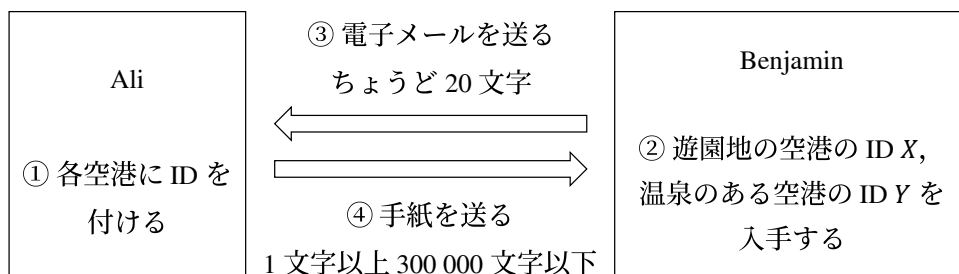
## 飛行機旅行 (Flights)

JOI 国には  $N$  個の空港があり、 $0$  から  $N-1$  までの番号が付いている。JOI 国には  $N-1$  本の航空路線があり、 $0$  から  $N-2$  までの番号が付いている。航空路線  $i$  ( $0 \leq i \leq N-2$ ) は空港  $U_i$  と空港  $V_i$  を双方向に結んでいる。どの空港からどの空港へも、いくつかの航空路線を乗り継ぐことで移動することができる。また、どの空港についても、その空港と他の空港を結ぶ航空路線は高々 3 本である。

Benjamin は JOI 国を旅行することを計画しており、最終日には遊園地のある空港から温泉のある空港に移動したい。遊園地のある空港に付けられた番号は  $x$  であり、温泉のある空港に付けられた番号は  $y$  である。しかし、彼は JOI 国の航空路線について何も知らないので、航空会社のスタッフである Ali と通信することで、遊園地のある空港から温泉のある空港まで移動するために最小何本の航空路線を利用する必要があるかを求めることにした。Ali は航空路線の情報を知っているが、Benjamin がどの空港の間を移動したいかを知らない。

1. Ali は各空港に ID を設定する。ID は  $0$  以上  $2N+19$  以下の整数でなければならない。
2. Benjamin は遊園地のある空港の ID  $X$  と、温泉のある空港の ID  $Y$  を入手する。
3. Benjamin は Ali に電子メールで  $0$  と  $1$  からなる文字列を伝える。文字列は ちょうど 20 文字 でなければならない。
4. Ali は Benjamin に手紙で  $0$  と  $1$  からなる文字列を伝える。文字列は  $1$  文字以上  $300\,000$  文字以下でなければならない。

航空会社のスタッフである Ali と、観光客である Benjamin の戦略を実装したプログラムを作成せよ。なお、手順 2. では Benjamin は遊園地・温泉のある空港の ID  $X, Y$  を入手することができるが、空港の番号  $x, y$  は入手できないことに注意せよ。





## 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 `Ali.cpp` という名前である。このファイルは `Ali` の戦略を実装したファイルであり、以下の2つ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 `Ali.h` を読み込むこと。

- `void Init(int N, std::vector<int> U, std::vector<int> V)`

これは、空港に `ID` を付ける `Ali` の戦略を実装した関数である。この関数は、各シナリオ（詳しくは採点の方法の項を参照）について、最初に1回だけ呼び出される。

- 引数 `N` は、`JOI` 国にある空港の数である。
- 引数 `U`, `V` は長さ  $N-1$  の配列であり、`U[i]`, `V[i]` は航空路線  $i$  が結んでいる空港  $U_i, V_i$  を表す ( $0 \leq i \leq N-2$ )。

- `std::string SendA(std::string S)`

これは、`Benjamin` に手紙を送る `Ali` の戦略を実装した関数である。この関数は、各シナリオ（詳しくは採点の方法の項を参照）について、関数 `SendB`（後述）が呼び出された後に1回だけ呼び出される。

- 引数 `S` は長さ20の文字列であり、受信した電子メールを表す。
- 関数 `SendA` は、送信する手紙を表す文字列を返さなければならない。
- 戻り値が長さ1以上300000以下の文字列ではない場合、不正解 [5] と判定される。
- 戻り値の文字列におけるすべての文字は `0`, `1` のいずれかでなければならず、これが満たされていない場合、不正解 [6] と判定される。

関数 `Init` の1回の呼び出しにおいて、以下の関数を、それぞれの空港に対して1回ずつ、合計  $N$  回呼び出さなければならない。

- `void SetID(int p, int value)`

- 引数 `p` は、`Ali` が空港 `p` の `ID` を設定することを表す。ここで、 $0 \leq p \leq N-1$  でなければならない。この範囲外の値で関数を呼び出した場合、不正解 [1] と判定される。
- 引数 `value` は、`Ali` が指定した空港に設定する `ID` である。ここで、 $0 \leq value \leq 2N+19$  でなければならない。この範囲外の値で関数を呼び出した場合、不正解 [2] と判定される。
- 関数 `SetID` を同じ `p` で2回呼び出した場合、不正解 [3] と判定される。
- 関数 `Init` の実行の終了時に関数 `SetID` の呼び出し回数が  $N$  回でなかった場合、不正解 [4] と判定される。

関数 `SetID` の呼び出しが不正解と判定された場合、その時点でプログラムは終了する。



2つ目のファイルは `Benjamin.cpp` という名前である。このファイルは Benjamin の戦略を実装したファイルであり、以下の2つ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 `Benjamin.h` を読み込むこと。

- `std::string SendB(int N, int X, int Y)`

これは、Ali に電子メールを送る Benjamin の戦略を実装した関数である。この関数は、各シナリオ（詳しくは採点の方法の項を参照）について、関数 `Init` が呼び出された後に1回だけ呼び出される。

- 引数 `N` は、JOI 国にある空港の数である。
- 引数 `X` は、遊園地のある空港の ID である。
- 引数 `Y` は、温泉のある空港の ID である。
- 関数 `SendB` は、送信する電子メールを表す文字列を返さなければならない。
- 戻り値が長さ 20 の文字列ではない場合、不正解 [7] と判定される。
- 戻り値の文字列におけるすべての文字は `0, 1` のいずれかでなければならず、これが満たされていない場合、不正解 [8] と判定される。

- `int Answer(std::string T)`

この関数では、空港 `x` から空港 `y` まで行くときに利用する航空路線の数の最小値を決定しなければならない。この関数は、各シナリオ（詳しくは採点の方法の項を参照）について、関数 `SendA` が呼び出された後に1回だけ呼び出される。

- 引数 `T` は長さ 1 以上 300 000 以下の文字列であり、受信した手紙を表す。
- この関数は、空港 `x` から空港 `y` まで行くときに利用する航空路線の数の最小値を返さなければならない。

## 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 Ali 側、Benjamin 側として2個のプロセスとして実行されるので、Ali 側と Benjamin 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。



## 採点の方法

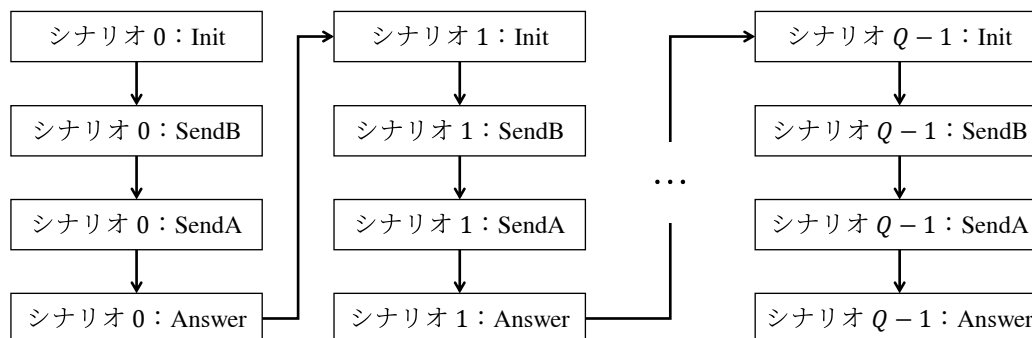
1つのテストケースは  $Q$  個のシナリオからなり、シナリオには 0 から  $Q-1$  までの番号が付けられている。各シナリオについて、以下の値が定められている。ただし、以下に記された値の範囲については、制約の項を参照せよ。

- JOI 国に存在する空港の数  $N$ .
- 遊園地のある空港の番号  $x$ .
- 温泉のある空港の番号  $y$ .
- 航空路線の情報  $(U_0, V_0), (U_1, V_1), \dots, (U_{N-2}, V_{N-2})$ .

関数 `Init`, `SendB`, `SendA`, `Answer` は各シナリオに対して実行され、与えられた引数に対して適切な関数を呼び出したり、適切な値を返したりしなければならない。ここで、以下のような手順でそれぞれの関数が呼び出される。

1.  $k = 0, 1, \dots, Q-1$  の順に、以下の処理 2 ~ 5 をこの順に行う。
2. 関数 `Init` が呼び出される。引数は、シナリオ  $k$  について実装の詳細の項に書かれた通りに設定される。
3. 関数 `SendB` が呼び出される。引数は、シナリオ  $k$  について実装の詳細の項に書かれた通りに設定される。
4. 関数 `SendA` が呼び出される。引数は、シナリオ  $k$  について実装の詳細の項に書かれた通りに設定される。
5. 関数 `Answer` が呼び出される。引数は、シナリオ  $k$  について実装の詳細の項に書かれた通りに設定される。

ただし、これらの処理の途中で不正解と判定された場合、その時点で採点は終了し、そのテストケースは不正解となる。





## 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`, `Ali.cpp`, `Benjamin.cpp`, `Ali.h`, `Benjamin.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++17 -O2 -o grader grader.cpp Ali.cpp Benjamin.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。ただし、入力はすべて整数でなければならない。

$Q$

(シナリオ 0 に対する入力)

(シナリオ 1 に対する入力)

⋮

(シナリオ  $Q-1$  に対する入力)

また、それぞれのシナリオに対する入力は、以下の形式で与えられる。

$N \ x \ y$

$U_0 \ V_0$

$U_1 \ V_1$

⋮

$U_{N-2} \ V_{N-2}$



## 採点プログラムのサンプルの出力

不正解 [1] から不正解 [8] までのいずれかであると判定された場合、不正解の種類が “Wrong Answer [1]” のように出力される（引用符は実際には出力されない）。

そうでない場合は、各シナリオに対する関数 `Answer` の戻り値と、Ali から Benjamin に送信された文字列の長さの最大値が次のように出力される。採点プログラムのサンプルは、関数 `Answer` の戻り値が正しいかどうかは判定しない。

```
Scenario 0: Your Answer = 3
Scenario 1: Your Answer = 1
Scenario 2: Your Answer = 4
Scenario 3: Your Answer = 1
Scenario 4: Your Answer = 5
Accepted: Maximum Length = 24
```

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。また、最初のシナリオで不正解と判定されなかった場合、その後のシナリオで不正解 [1] から不正解 [8] までのいずれかであると判定されても、次のように途中までの結果が出力されることがある。

```
Scenario 0: Your Answer = 3
Scenario 1: Your Answer = 1
Scenario 2: Your Answer = 4
Wrong Answer [8]
```

## 制約

- $1 \leq Q \leq 50$ .
- $2 \leq N \leq 10\,000$ .
- $0 \leq U_i < V_i \leq N - 1$  ( $0 \leq i \leq N - 2$ ).
- $0 \leq x \leq N - 1$ .
- $0 \leq y \leq N - 1$ .
- $x \neq y$ .
- どの空港からどの空港へも、いくつかの航空路線を乗り継ぐことで移動することができる。
- どの空港についても、その空港と他の空港を結ぶ航空路線は高々 3 本である。



## 小課題

1. (15 点)  $Q = 1$ .
2. (85 点)  $Q \geq 2$ .

### 小課題 1 の採点基準

小課題 1 に対応するテストケースについて、1 つでも不正解があった場合、この小課題の得点は 0 点となる。

また、小課題 1 に対応するすべてのテストケースに正解した場合、小課題 1 のすべてのテストケースに対する「Ali から Benjamin へ送信された文字列の長さの最大値」を  $L_1$  として、この小課題の得点は以下のように与えられる。

$L_1$ の値	得点
$150\,001 \leq L_1 \leq 300\,000$	7 点
$20\,001 \leq L_1 \leq 150\,000$	11 点
$L_1 \leq 20\,000$	15 点

### 小課題 2 の採点基準

小課題 2 に対応するテストケースについて、1 つでも不正解があった場合、この小課題の得点は 0 点となる。

また、小課題 2 に対応するすべてのテストケースに正解した場合、小課題 2 のすべてのテストケースに対する「Ali から Benjamin へ送信された文字列の長さの最大値」を  $L_2$  として、この小課題の得点は以下のように与えられる。特に  $1401 \leq L_2$  であれば、この小課題の得点が 0 点となることに注意せよ。

$L_2$ の値	得点
$1401 \leq L_2 \leq 300\,000$	0 点
$71 \leq L_2 \leq 1400$	$52 - 35 \times \log_{10} \left( \frac{L_2}{70} \right)$ 点を整数に切り捨てた得点
$45 \leq L_2 \leq 70$	$87 - 0.5 \times L_2$ 点を整数に切り捨てた得点
$25 \leq L_2 \leq 44$	$109 - L_2$ 点
$L_2 \leq 24$	85 点



## 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。ただし以下の例は、Ali が関数 `Init` で、空港 0, 1, 2, 3 にそれぞれ 12, 21, 25, 27 という ID を設定した場合である。

入力例 1
1
4 0 2
0 1
1 2
2 3

Ali の呼び出し	Ali の戻り値	Benjamin の呼び出し	Benjamin の戻り値
<code>Init(4, [0, 1, 2], [1, 2, 3])</code>			
<code>SetID(0, 12)</code>			
<code>SetID(1, 21)</code>			
<code>SetID(2, 25)</code>			
<code>SetID(3, 27)</code>			
		<code>SendB(12, 25)</code>	<code>"00000111110000011111"</code>
<code>SendA("00...11")</code>	<code>"10"</code>		
		<code>Answer("10")</code>	2

この例では空港が  $N (= 4)$  個あり、次の 3 つの航空路線が存在する。

- 空港 0 と空港 1 を結ぶ航空路線。
- 空港 1 と空港 2 を結ぶ航空路線。
- 空港 2 と空港 3 を結ぶ航空路線。

空港  $x (= 0)$  から空港  $y (= 2)$  まで行くためには最小で 2 本の航空路線を利用する必要があるため、関数 `Answer` は 2 を返さなければならない。

なお、関数 `SendB` の引数は空港の番号  $(x, y) = (0, 2)$  ではなく、空港の ID  $(X, Y) = (12, 25)$  であることに注意すること。





入力例 2

```
2
10 0 9
0 1
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
15 12 8
0 1
0 2
1 3
1 4
2 5
2 6
3 7
3 8
4 9
4 10
5 11
5 12
6 13
6 14
```

この例では、 $Q = 2$  個のシナリオが存在する。

- 1 個目のシナリオでは、関数 **Answer** は 9 を返さなければならない。
- 2 個目のシナリオでは、関数 **Answer** は 6 を返さなければならない。