



危険なスケート (Dangerous Skating)

JOI 君は大自然の広大なスケートリンクでアイススケートをするのが趣味である。

スケートリンクは南北に R マス、東西に C マスからなる長方形で表される。北から r 行目、西から c 列目のマスをマス (r, c) と書くことにする。それぞれのマスは JOI 君が通過可能なマスであるか、氷塊があって通過不可能なマスであるかのいずれかである。また、スケートリンクの外周のマスにはすべて氷塊があり、リンクの外に出てしまうことはない。すなわち、マス $(i, 1), (i, C)$ ($1 \leq i \leq R$) およびマス $(1, j), (R, j)$ ($1 \leq j \leq C$) には氷塊がある。

JOI 君はあまりスケートがうまくない。JOI 君はスケートリンク上で移動するときは、東西南北のうち 1 つの方向に向かってリンクの現在 JOI 君がいるマスを蹴り、氷塊にぶつかる直前のマスまで進んで止まる。リンクを蹴ってから止まるまでを 1 回の移動と数える。隣接するマスに氷塊があるときは、その方向には移動できない。

ある日 JOI 君がスケートを楽しんでいると、なんと、JOI 君がリンクを蹴ると、そのマスに氷塊が生えてくることに気がついた。リンクを蹴った地点以外で通過したマスには氷塊は生えない。この状態でスケートを続けるのは非常に危険なので、JOI 君はできるだけ早くこのスケートリンクから脱出したい。

JOI 君の現在地はマス (r_1, c_1) である。このスケートリンクから脱出するには、出口のマス (r_2, c_2) で止まる必要がある。JOI 君が安全にスケートリンクから脱出できるよう、現在地から移動を開始して出口のマスで止まるには、少なくとも何回の移動が必要かを計算するプログラムを書いてほしい。スケートリンクの状態や JOI 君の現在地によっては、JOI 君がどのように移動しても出口のマスに止まることができないこともあるかもしれない。JOI 君が移動の途中に出口のマスを通じただけでは、スケートリンクからは脱出できないことに注意せよ。

課題

スケートリンク上の氷塊の情報と、JOI 君の現在地、出口のマスの位置が与えられたとき、JOI 君が現在地から移動を開始して出口のマスで止まることができるかどうかを判定し、できる場合は、そのために必要な移動回数の最小値を求めるプログラムを作成せよ。

入力

標準入力から以下のデータを読み込め。

- 1 行目には、整数 R, C が空白を区切りとして書かれている。これは、スケートリンクの大きさが南北に R マス、東西に C マスであることを表す。
- 続く R 行のそれぞれには、 C 文字からなる文字列が書かれている。各文字は '.' または '#' のいずれかである。この R 行のうちの r 行目 ($1 \leq r \leq R$) の左から c 文字目 ($1 \leq c \leq C$) は、スケートリンクのマス (r, c) の初期状態を表す。この文字が '.' のときは、そのマスが通行可能であることを表し、この文字が '#' のときは、そのマスに氷塊があって通行不可能であることを表す。



- 続く 1 行には、整数 r_1, c_1 が空白を区切りとして書かれている。これは、JOI 君の現在地がマス (r_1, c_1) であることを表す。
- 続く 1 行には、整数 r_2, c_2 が空白を区切りとして書かれている。これは、スケートリンクの出口がマス (r_2, c_2) であることを表す。

出力

標準出力に、JOI 君が現在地から移動を開始して出口のマスで止まるために必要な移動回数の最小値を表す整数を 1 行で出力せよ。ただし、JOI 君がどのように移動しても出口のマスで止まることができない場合は、-1 を出力せよ。

制限

すべての入力データは以下の条件を満たす。

- $3 \leq R \leq 1000$.
- $3 \leq C \leq 1000$.
- $1 \leq r_1 \leq R$.
- $1 \leq c_1 \leq C$.
- $1 \leq r_2 \leq R$.
- $1 \leq c_2 \leq C$.
- スケートリンクの外周のマスにはすべて氷塊がある。すなわち、マス $(i, 1), (i, C)$ ($1 \leq i \leq R$) およびマス $(1, j), (R, j)$ ($1 \leq j \leq C$) には氷塊がある。
- マス (r_1, c_1) およびマス (r_2, c_2) に氷塊はない。

小課題

小課題 1 [13 点]

以下の条件を満たす。

- $R \leq 10$.
- $C \leq 10$.
- JOI 君が現在地から移動を開始して出口のマスで止まることができる場合は、必要な移動回数は 10 回以内である。



小課題 2 [65 点]

以下の条件を満たす.

- $R \leq 200$.
- $C \leq 200$.

小課題 3 [22 点]

追加の制限はない.

入出力例

入力例 1	出力例 1
5 5 ##### #...# #...# #...# ##### 2 2 3 3	4

入力例 1 において, スケートリンクの初期状態は以下の通りである. 白の四角形が書かれたマスは氷塊を, J と書かれたマスは JOI 君の現在地を, E と書かれたマスは出口を表す.

□	□	□	□	□
□	J			□
□		E		□
□				□
□	□	□	□	□

まず, JOI 君が東向きに移動すると, その後のスケートリンクの状態は以下のようになる.

□	□	□	□	□
□	□		J	□
□		E		□
□				□
□	□	□	□	□

この後 JOI 君が西向き, 南向き, 北向きの順に移動することで, 合計 4 回の移動で出口で止まることができる. 3 回以下の移動で出口で止まることができないので, 4 を出力する.



入力例 2	出力例 2
8 6 ##### #.#.# ##...# #....# #.#..# #....# ##...# ##### 4 3 6 4	5

入力例 3	出力例 3
5 5 ##### #.#.# #.#.# #.#.# ##### 2 2 4 4	-1

入力例 4	出力例 4
3 3 ### #.# ### 2 2 2 2	0

入力例 4 において、JOI 君の現在地は出口のマスであるから、必要な移動回数は 0 回である。



雪降る道路 (Snowy Roads)

ロシアの多くの地点では冬季に雪が降る。ロシアには 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。ロシアには $N - 1$ 本の道路があり、 0 から $N - 2$ までの番号が付けられている。道路 i ($0 \leq i \leq N - 2$) は、2つの異なる都市 A_i, B_i ($0 \leq A_i < B_i \leq N - 1$) を双方向に結んでいる。ロシアのどの2つの異なる都市の間も、いくつかの道路を経由して行き来可能になっている。

各道路の降雪状況は日によって変化する。どの日付においても、各道路の降雪状況は、雪が降っているか、降っていないかのいずれかである。1日のうちに降雪状況が変化することはない。

Anya と Boris はロシアの交通局で働いている。Anya は道路情報の管理部門、Boris は市民からの質問に回答する部門に所属している。市民からの質問は、ロシアの首都である都市 0 から他のある都市まで移動するためには、雪が降っている道路を最低でも何本通らなければならないか、というものである。Boris は通常、質問を受けた後に Anya とやり取りをして、市民に回答する。

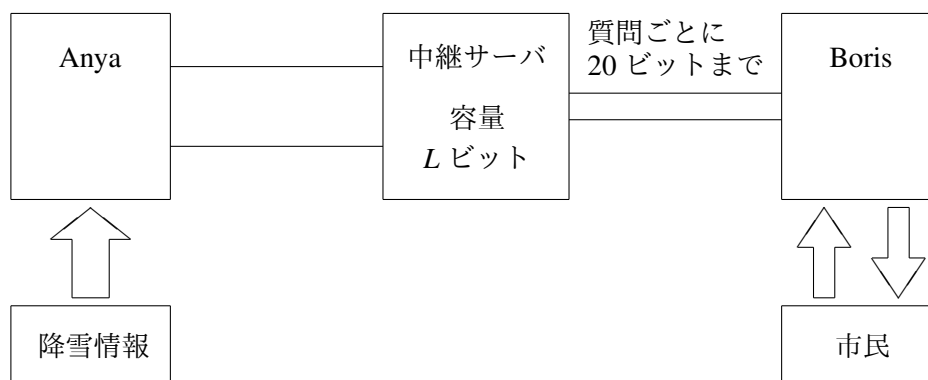
この度、ロシアでは Q 日間に渡りプログラミングコンテストの世界大会が開催されることとなった。大会期間中は、通信回線が混雑し、Anya と Boris が直接やり取りをすることが難しくなると予想される。そこで、Anya と Boris は、次の方法でやり取りをすることになった。

- Anya は、1日の始めにその日の降雪情報を受け取り、中継サーバにデータを送る。
- Boris は、市民からの質問を受けると、中継サーバとやり取りをすることによって回答する。

ただし、Anya と Boris のやり取りには次の制約がある。

- 中継サーバの容量は $L = 1000$ ビットである。Anya は中継サーバに最大で L ビットの情報しか格納できない。
- 中継サーバに格納されているデータは、1日の始めにすべて 0 に初期化される。
- Boris は、中継サーバとの1回のやり取りにより、指定した1ビットの情報を読み出すことができる。
- 質問への回答のために Boris が中継サーバとやり取りできるのは、各質問ごとに高々20回までである。

交通局長と知り合いのあなたは、Anya と Boris の戦略を考えることとなった。





課題

市民からの質問に正しく回答できる Anya と Boris の戦略を実装したプログラムを作成せよ。

実装の詳細

あなたは同じプログラミング言語で 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 `Anya.c` または `Anya.cpp` という名前である。このファイルは Anya の戦略を実装したファイルであり、以下の 2 つのルーチンを実装していなければならない。プログラムは `Anyalib.h` をインクルードすること。

- `void InitAnya(int N, int A[], 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` は都市の個数を表す。
- 引数 `A[]` と `B[]` はそれぞれ長さ $N - 1$ の数列であり、道路の接続情報を表す。要素 `A[i]` および要素 `B[i]` ($0 \leq i \leq N - 2$) は道路 i が都市 `A[i]` と `B[i]` を双方向に結んでいることを表す整数であり、 $0 \leq A[i] < B[i] \leq N - 1$ を満たす。

- `void Anya(int C[])`

この関数は `InitAnya` が呼び出されたあと、 Q 回呼び出される。この関数は、1 日の始めに道路の降雪情報が更新されたあとに、Anya が中継サーバに保存するビット列を決めることに対応する。

- 引数 `C[]` は長さ $N - 1$ の数列であり、道路の降雪情報を表す。要素 `C[i]` ($0 \leq i \leq N - 2$) は道路 i の降雪情報を表す 0 か 1 の整数であり、`C[i] = 1` なら道路 i に雪が降っていることを、`C[i] = 0` なら道路 i に雪が降っていないことを表す。

関数 `Anya` 中では以下の関数を呼び出すことができる。

- ★ `void Save(int place, int bit)`

この関数は Anya が中継サーバにビットを保存する操作を表す。

- 引数 `place` はビットを書き込む場所を表す。`place` は 0 以上 $L - 1$ 以下の整数でなければならない。この範囲外の値を指定して呼び出した場合、不正解 [1] と判定される。また、関数 `Anya` の各呼び出しにおいて、同じ引数 `place` で 2 回以上呼び出すことはできない。同じ引数で 2 回呼び出した場合、不正解 [2] と判定される。
- 引数 `bit` は書き込むビットを表す整数であり、0 か 1 でなければならない。これ以外の値を指定して呼び出した場合、不正解 [3] と判定される。



Save の呼び出し後、中継サーバの第 `place` ビットの中身は、`bit` になる。

Save 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。

関数 `Anya` が呼び出される直前に、必ず、中継サーバのビットはすべて 0 に初期化される。すなわち、関数 `Save` による保存操作が行われなかった場所には、関数 `Anya` 終了時には 0 が書き込まれている。

2 つ目のファイルは `Boris.c` または `Boris.cpp` という名前である。このファイルは `Boris` の戦略を実装したファイルであり、以下の 2 つのルーチンを実装していなければならない。プログラムは `Borislib.h` をインクルードすること。

- `void InitBoris(int N, int A[], 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` は都市の個数を表す。
- 引数 `A[]` と `B[]` はそれぞれ長さ $N - 1$ の数列であり、道路の接続情報を表す。要素 `A[i]` および要素 `B[i]` ($0 \leq i \leq N - 2$) は道路 i が都市 `A[i]` と `B[i]` を双方向に結んでいることを表す整数であり、 $0 \leq A[i] < B[i] \leq N - 1$ を満たす。

- `int Boris(int city)`

この関数は `InitBoris` が呼び出されたあとに何回か呼び出される。この関数は市民の質問に対する `Boris` の行動に対応する。

- 引数 `city` は市民の質問を表す。`city` は 1 以上 $N - 1$ 以下の整数である。これは都市 0 から都市 `city` まで移動するためには、雪が降っている道路を最低でも何本通らなければならないかを、市民が質問していることを表す。
- 関数 `Boris` は、市民の質問への回答にあたる 0 以上 $N - 1$ 以下の整数を返さなければならない。この範囲外の整数を返した場合、不正解 [4] と判定される。回答が正しくない場合、不正解 [7] と判定される。

関数 `Boris` 中では以下の関数を呼び出すことができる。

- ★ `int Ask(int place)`

この関数は `Boris` が中継サーバからビットを読み出す操作を表す。

- 引数 `place` はビットを読み出す場所を表す。`place` は 0 以上 $L - 1$ 以下の整数でなければならない。この範囲外の値を指定して呼び出した場合、不正解 [5] と判定される。

関数 `Ask` の戻り値は、中継サーバの第 `place` ビットの中身を表す整数であり、0 または 1 である。また、関数 `Ask` は、関数 `Boris` の各呼び出しにおいて、高々 20 回しか呼び出すことはできない。20 回を超えて呼び出した場合、不正解 [6] と判定される。

`Ask` 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。



採点の手順

採点は以下の手順で行われる。不正解と判定された場合はその時点でプログラムは終了される。

- (1) 道路情報を与えるために、`InitAnya` と `InitBoris` が 1 回ずつこの順番に呼び出される。
- (2) $i = 1, \dots, Q$ に対して、順番に以下の操作を行う。
 - (a) 関数 `Anya` を 1 回だけ呼び出す。これは、1 日の始めに道路の降雪情報が更新されたあとに、`Anya` が中継サーバに保存するビット列を決めることに対応する。
 - (b) 関数 `Boris` を D_i 回呼び出す。 D_i は i 日目における市民からの質問の回数である。このうち j 回目 ($1 \leq j \leq D_i$) の呼び出しにおける引数は R_{ij} ($1 \leq R_{ij} \leq N - 1$) である。 j 回目の呼び出しにおいて、関数 `Boris` の戻り値が、都市 0 から都市 R_{ij} まで移動するために通る必要のある、雪が降っている道路の本数の最小値と一致していない場合、不正解となる。
- (3) 一度も不正解と判定されなかった場合、正解となる。

重要な注意

- 実行時間計測・使用メモリ計測の対象となるのは、「採点の手順」における手順 (1), (2) である。正解の場合は、手順 (2) において、`Anya` は合計 Q 回、`Boris` は合計 $D_1 + \dots + D_Q$ 回呼び出されることになる。
- `Anya` や `Boris` は、 D_1, \dots, D_Q の値を知ることはできない。
- 関数 `Boris` には、どの時点で関数 `Anya` が中継サーバに保存されているデータを更新したかを表す情報は与えられない。
- 内部での使用のために他のルーチンを実装したり、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ルーチン (`InitAnya`, `Anya`, `InitBoris`, `Boris` を除く) を `static` 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 `Anya` 側、`Boris` 側として 2 個のプロセスとして実行されるので、`Anya` 側と `Boris` 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。

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.c` または `grader.cpp` である。作成したプログラムを `Anya.c` および `Boris.c`、または `Anya.cpp` および `Boris.cpp` とするとき、作成したプログラムをテストするには、次のようにコマンドを実行する。

- C の場合

```
gcc -std=c11 -O2 -o grader grader.c Anya.c Boris.c -lm
```

- C++ の場合

```
g++ -std=c++11 -O2 -o grader grader.cpp Anya.cpp Boris.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概要

採点プログラムのサンプルは、「採点の手順」に従って関数を呼び出す。

以下の点は実際の採点プログラムの挙動とは異なるので注意せよ。

- 採点プログラムのサンプルは、不正解 [7] の判定を行わず、代わりに各質問に対する関数 `Boris` の戻り値を出力する。

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データを読み込む。

- 1 行目には整数 N が書かれている。これは都市の数が N 個であることを表す。
- 続く $N-1$ 行のうちの $i+1$ 行目 ($0 \leq i \leq N-2$) には、整数 A_i, B_i が空白を区切りとして書かれている。これは道路 i が都市 A_i と都市 B_i を双方向に結んでいることを表す。
- 次の行には整数 Q が書かれている。これは大会が Q 日間に渡り開催されることを表す。
- 続く Q 行のうちの i 行目 ($1 \leq i \leq Q$) には、長さ $N-1$ の文字列 S_i と D_i+1 個の整数が空白を区切りとして書かれている。 S_i は i 日目の降雪情報を表しており、 S_i の左から $j+1$ 文字目 ($0 \leq j \leq N-2$) が '1' なら道路 j に雪が降っていることを、 '0' なら道路 j に雪が降っていないことを表す。続く D_i+1 個の整数のうちの先頭の整数は、 D_i である。続く D_i 個の整数は、 R_{i1}, \dots, R_{iD_i} である。これは、 i 日目において、 j 番目 ($1 \leq j \leq D_i$) の質問が、都市 0 から都市 R_{ij} まで移動するために通る必要のある、雪が降っている道路の本数の最小値に関するものであることを表す。



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）。

- プログラムの実行中に不正解と判定された場合、不正解の種類が“Wrong Answer [1]”のように出力され、実行が終了される。
- i 番目 ($1 \leq i \leq Q$) の Anya の呼び出し後に呼び出される D_i 個の Boris の呼び出しすべてにおいて不正解と判定されなかった場合、その時点で 1 行に D_i 個の整数が空白を区切りとして出力される。出力された整数のうちの j 番目 ($1 \leq j \leq D_i$) の整数は、 $\text{Boris}(R_{ij})$ の戻り値である。

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。

制限

すべての入力データは以下の条件を満たす。

- $2 \leq N \leq 500$.
- $1 \leq Q \leq 500$.
- $0 \leq A_i < B_i \leq N - 1$ ($0 \leq i \leq N - 2$).
- $1 \leq D_j$ ($1 \leq j \leq Q$).
- $D_1 + \dots + D_Q \leq 500$.
- どの 2 つの異なる都市の間も、いくつかの道路を経由して行き来可能である。

小課題

小課題 1 [15 点]

- $N \leq 20$ を満たす。

小課題 2 [5 点]

- $N \leq 100$ を満たす。



小課題 3 [35 点]

以下の条件を満たす.

- $A_i = i$ ($0 \leq i \leq N - 2$).
- $B_i = i + 1$ ($0 \leq i \leq N - 2$).

小課題 4 [45 点]

追加の制限はない.



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	ルーチンの呼び出しの例			
	呼び出し	戻り値	呼び出し	戻り値
5	InitAnya(...)			
0 1		(なし)		
1 2	InitBoris(...)			
1 4		(なし)		
2 3	Anya(...)			
2			Save(0,1)	
0101 3 2 4 3				(なし)
1110 1 4			Save(1,0)	
				(なし)
			Save(500,1)	
				(なし)
		(なし)		
	Boris(2)			
			Ask(0)	
				1
			Ask(1)	
				0
			Ask(2)	
				0
		1		
	Boris(4)			
		0		
	Boris(3)			
		2		
	Anya(...)			
		(なし)		
	Boris(4)			
			Ask(0)	
				0
		2		

この例での関数の呼び出しは、必ずしも意味のある呼び出しとは限らないことに注意せよ。

このとき、InitAnya(...), InitBoris(...), 1回目および2回目の Anya(...) に渡される引数はそれぞれ次の通りである。



引数	InitAnya(...)	InitBoris(...)	Anya(...)	
			1 回目	2 回目
N	5	5		
A	{0, 1, 1, 2}	{0, 1, 1, 2}		
B	{1, 2, 4, 3}	{1, 2, 4, 3}		
C			{0, 1, 0, 1}	{1, 1, 1, 0}



最悪の記者 2 (Worst Reporter 2)

時は 21XX 年，競技プログラミングはマインドスポーツの 1 つとして広く認知されており，テレビ，新聞などのメディアで取り上げられることも多い。

あなたは JOI 新聞社の記者であり，競技プログラミングの記事を担当している。

昨日， N 人の選手による国際的な競技プログラミングのコンテストが開催された。このコンテストについての記事を書くために，あなたには次の情報が与えられた。

- 国際情報オリンピックなどと同様，このコンテストにはいくつかの国から選手が参加した。国には 1 から N までのいずれかの番号が付けられている。一つの国から複数の選手が参加することもあり得る。また，選手が参加しない国があるかもしれない。
- このコンテストの競技時間は 5 時間である。
- コンテスト中に選手の獲得した点数が，その後減らされることはない。
- コンテスト開始後 2 時間経過した時点において，同点の選手はいなかった。その時点の順位表において， i 位 ($1 \leq i \leq N$) の選手は国 A_i の出身で，その選手の点数は B_i 点であった。
- コンテスト終了時点において，同点の選手はいなかった。コンテスト終了時点の順位表において， i 位 ($1 \leq i \leq N$) の選手は国 C_i の出身で，その選手の点数は D_i 点であった。

しかしながら，記事を書く段階になって，順位表の出身国の表示に不具合があったことが判明した。選手の出身国の情報が間違っ表示されていた可能性がある。表示されていた選手の点数は正しいことが分かっている。

そこで，あなたは，与えられた情報にできるだけ少ない修正を加えることで，順位表の情報として矛盾のない(同じ選手の出身国がコンテスト中に変わったり，選手の獲得した点数がコンテスト中に減少したりしない)ものを推測することにした。すなわち， $2N$ 個の値 $A_1, \dots, A_N, C_1, \dots, C_N$ のうちでできるだけ少ない箇所を変更することで，次の条件を満たすようにしたい：

- $1, 2, \dots, N$ のある並び替え x_1, x_2, \dots, x_N であって，各 $i = 1, 2, \dots, N$ に対して $A_i = C_{x_i}$ かつ $B_i \leq D_{x_i}$ が成り立つものが存在する。

あなたは，与えられた情報に，最少で何箇所の修正を加える必要があるだろうか。

課題

コンテストの参加者数と，コンテスト開始後 2 時間経過した時点とコンテスト終了時点の順位表についての情報が与えられたとき，順位表を矛盾のない状態にするために必要な，出身国情報の変更箇所の個数の最小値を求めるプログラムを作成せよ。



入力

標準入力から以下のデータを読み込め。

- 1行目には、整数 N が書かれている。これは、コンテストに N 人の選手が参加したことを表す。
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には、整数 A_i, B_i が空白を区切りとして書かれている。これは、コンテスト開始後2時間経過した時点の順位表において、 i 位の選手は国 A_i 出身と表示され、獲得した点数は B_i 点であったことを表す。
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には、整数 C_i, D_i が空白を区切りとして書かれている。これは、コンテスト終了時点の順位表において、 i 位の選手は国 C_i 出身と表示され、獲得した点数は D_i 点であったことを表す。

出力

標準出力に、順位表を矛盾のない状態にするために必要な、出身国情報の変更箇所の個数の最小値を1行で出力せよ。

制限

すべての入力データは以下の条件を満たす。

- $2 \leq N \leq 200\,000$.
- $1 \leq A_i \leq N$ ($1 \leq i \leq N$).
- $0 \leq B_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).
- $B_i > B_{i+1}$ ($1 \leq i \leq N - 1$).
- $1 \leq C_i \leq N$ ($1 \leq i \leq N$).
- $0 \leq D_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).
- $D_i > D_{i+1}$ ($1 \leq i \leq N - 1$).
- $A_1, \dots, A_N, C_1, \dots, C_N$ の値を何箇所か変更することにより、順位表を矛盾のない状態にすることができる。

小課題

小課題 1 [15 点]

- $N \leq 16$ を満たす。



小課題 2 [15 点]

- $N \leq 50$ を満たす.

小課題 3 [30 点]

- $N \leq 5000$ を満たす.

小課題 4 [40 点]

追加の制約はない.

入出力例

入力例 1	出力例 1
3	1
3 500	
2 200	
1 100	
1 1000	
3 700	
3 400	

C_3 の値を 2 に修正すると、次のように矛盾のない順位表になる：

- コンテスト開始後 2 時間経過した時点において 500 点で 1 位だった国 3 出身の選手は、コンテスト終了時点において 700 点で 2 位となった.
- コンテスト開始後 2 時間経過した時点において 200 点で 2 位だった国 2 出身の選手は、コンテスト終了時点において 400 点で 3 位となった.
- コンテスト開始後 2 時間経過した時点において 100 点で 3 位だった国 1 出身の選手は、コンテスト終了時点において 1000 点で 1 位となった.

ここで、 C_2 の値を 2 に修正した場合、国 3 出身の選手がコンテスト開始後 2 時間経過した時点において 500 点を獲得しているにも関わらず、コンテスト終了時点において 400 点となっているため、矛盾した順位表となる.

1 箇所より少ない修正によって矛盾のない順位表を得ることは不可能であるから、1 を出力する.



入力例 2	出力例 2
3	0
3 3	
3 2	
1 1	
3 4	
3 2	
1 1	

この場合、出身国情報を修正しなくても矛盾のない順位表となっている。コンテスト開始後2時間経過した時点の点数から点数を増やしていない選手が存在するかもしれないことに注意せよ。また、順位表において、同じ出身国の選手が複数いる可能性があることに注意せよ。

入力例 3	出力例 3
6	3
1 70	
4 50	
1 30	
2 20	
1 10	
3 0	
6 100	
2 90	
1 80	
2 60	
4 40	
1 10	

この入力例において、 A_1 の値を 2 に修正し、 A_6 の値を 4 に修正し、 C_1 の値を 4 に修正すると、矛盾のない順位表になる。