

第12回 日本情報オリンピック本選 課題¹

2013年2月10日

競技参加者への注意事項

- 競技開始の指示のあるまで、課題が入った封筒を開けないで下さい。
- 課題は5問、競技時間は4時間（10時～14時）です。
- 質問がある場合は、競技開始から2時間経過するまで質問票を提出することができます。質問への回答は「正しい」「正しくない」「ノーコメント」「課題の記述の中に答えはある」「無効」のいずれかです。競技開始から2時間経過後も質問票を提出することはできますが、競技時間内に回答するとは限りません。
- 実行時間制限・メモリ制限・コンパイラオプション・配点・応答の種別は **overview sheet** に記述されています。
- 全ての課題でセットごとに採点を行います。各セットは1個もしくは複数のデータで構成されており、セット中のすべてのデータに正解した場合にのみ配点分の得点が与えられます。
- 解答は、解答提出サイト <http://joi/> から提出して下さい。
- 提出した解答にはフィードバック（「完全」「部分」「例のみ」のいずれか）が与えられます。フィードバックが与えられるまでに時間がかかることがあります。競技終了15分前までに提出されたソースには競技時間内にフィードバックが与えられますが、それ以降に提出されたソースには競技時間内にフィードバックが与えられるとは限りません。ソースを提出する回数に制限はありません。複数回提出した場合は、提出された全てのソースが採点され、提出されたソースの得点の最大値がその問題の得点となります。
- **joi2013.zip** を <http://joi/> からダウンロードしてください。home directory で **joi2013.zip** を展開してください。home directory に **joi2013** という directory が作成され、その下に **2013-ho-t1**, **2013-ho-t2**, **2013-ho-t3**, **2013-ho-t4**, **2013-ho-t5** の5つの directory が作成されます。問題ごとの directory の中には問題文中の入出力例が展開されます。解答ソースは問題ごとの directory の中に、overview sheet に記載のファイル名で保存して下さい。競技サーバに予期せぬ障害が発生した場合に、これらの directory にあるファイルで採点を行う場合があります。

情報オリンピック日本委員会

¹Copyright ©2013 The Japanese Committee for International Olympiad in Informatics
著作権は情報オリンピック日本委員会に帰属します。



1

電飾 (Illumination)

JOI 高校の文化祭では毎年廊下に電飾が飾られる。電飾は N 個の電球で構成されており、電球は廊下の西側から東側に一列に並んでいる。各電球は明かりがついているか、ついていないかのいずれかの状態である。

JOI 高校の倉庫には電球を操作する機械が眠っている。この機械は電飾内で連続した電球を指定すると、指定された電球のうち、明かりがついている電球全てを明かりがついていない状態にし、明かりがついていない電球全てを明かりがついている状態にする。ただし、機械は老朽化のため、1 回しか使用できない。

JOI 高校の生徒達は明かりがついている電球とついていない電球が交互に並んだ列（このような電球の列を交互列と呼ぶ）が好きである。そこで、この機械を必要ならば 1 回だけ使って、できるだけ長い交互列を含む電飾を作ることにした。

例

例えば、電飾の配置が西から東に向かって

○ ○ ● ● ○ ● ○ ○ ○ ●

となっていたとする（○は明かりがついている電球を、●は明かりがついていない電球を表す）。このとき、4 番目から 7 番目までの 4 個の電球に対して機械を操作すると、

○ ○ ● ○ ● ○ ● ○ ○ ●

となり、2 番目から 8 番目までの電球が長さ 7 の交互列をなす。

○ ○ ● ○ ● ○ ● ○ ○ ●

また、8 番目の電球のみに対して機械を操作すると、

○ ○ ● ● ○ ● ○ ● ○ ●

となり、4 番目から 10 番目までの電球が長さ 7 の交互列をなす。

○ ○ ● ● ○ ● ○ ● ○ ●

機械を最大 1 回使用することで、長さが 8 以上の交互列を作ることはできない。

課題

電飾の情報が与えられたとき、機械を最大 1 回使用することで得られる電球の配列に含まれる交互列の長さとして考えられるものの最大値を求めるプログラムを作成せよ。



制限

$2 \leq N \leq 100\,000$ 電飾を構成する電球の個数

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.
- 2 行目には N 個の 0 または 1 が空白を区切りとして書かれている. 各整数は機械を操作する前における電球の情報を表している. 左から i ($1 \leq i \leq N$) 番目の整数は西側から i 番目の電球の情報を表しており, 整数が 1 ならば電球の明かりがついていて, 0 ならば明かりがついていないことを表す.

出力

標準出力に, 作成可能な電球の列に含まれる交互列の長さの最大値を表す整数を 1 行で出力せよ.

採点基準

採点用データのうち, 配点の 20%分については, $N \leq 500$ を満たす.
採点用データのうち, 配点の 40%分については, $N \leq 2\,000$ を満たす.



入出力例

入力例 1	出力例 1
10 1 1 0 0 1 0 1 1 1 0	7

これは問題文中で説明された例である。

入力例 2	出力例 2
10 1 0 0 0 0 1 0 1 0 1	8

西側から 4 番目の電球のみを操作すると、最大値 8 を満たす交互列が得られる。

入力例 3	出力例 3
5 1 1 0 1 1	5

西側から数えて 2 番目から 4 番目までの電球を操作すると、全ての電球からなる交互列を作ることができる。

入力例 4	出力例 4
3 0 1 0	3

機械を使用しなくても良い場合があることに注意せよ。

2

IOI 列車で行こう (Take the 'IOI' train)

IOI 国ではこのたび新たに鉄道を敷設した。IOI 国の鉄道を走る列車はいくつかの車両が連結されたものであり、車両には I, O の 2 種類がある。車両はそれぞれ異なる種類の車両としか連結できない。また、列車に運転席を設ける関係上、列車の両端の車両は種類 I でなければならない。列車は車両の種類を表す文字を順につなげた文字列で表され、列車の長さはその文字列の長さであるとする。たとえば、IOIOI の順に車両を連結すると長さ 5 の列車を編成でき、また車両 I は単独で長さ 1 の列車である。車両を OIOI や IOOI といった順に並べても列車を編成することはできない。

いくつかの車両が 2 つの車庫に格納されている。それぞれの車庫の中には車両が一行に並んでいる。列車を編成するときは車庫から車両を出してきて車庫前で連結していく。車庫から出せる車両は最も車庫の入り口に近い車両のみであるが、どちらの車庫から車両を出すかの順番については自由である。

列車を編成する前に、車両を好きだけ車庫から出して別の待機用レールに移すことができる。一度待機用レールに移した車両は今後列車を編成するために使うことはできない。また、一度列車の編成を始めるとその編成が終わるまでの間は車両を車庫から待機用レールに移すことはできない。

列車を編成するとき、車庫内の全ての車両を使い切る必要はない。すなわち、列車の編成を終えた後、車庫内に使われなかった車両が残っていても構わない。

IOI 国では鉄道に乗る人がとてもたくさんいると考えられているので、できるだけ長い列車を編成したい。

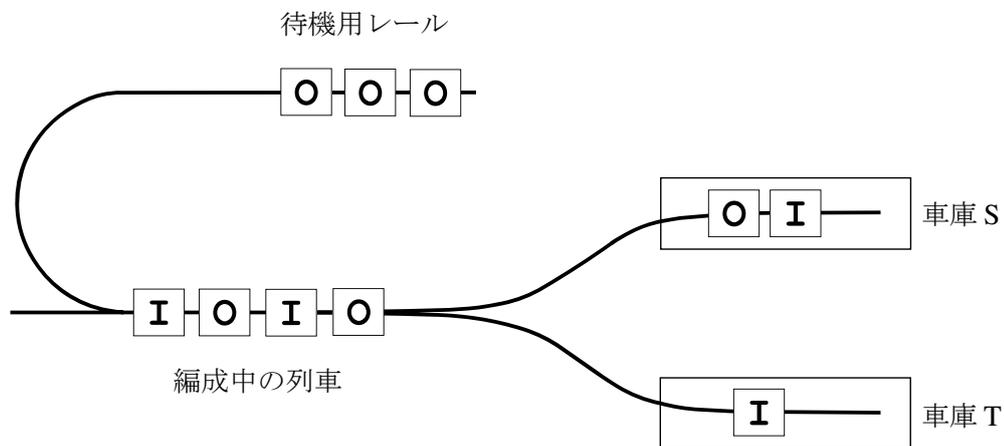


図: 列車を編成している途中であり、このとき車庫にある車両を待機用レールに移すことはできない。
この図は入出力例 1 に対応している。



課題

車庫に格納された車両の情報が与えられたとき，編成できる列車の長さの最大値を求めるプログラムを作成せよ．それぞれの車庫に格納された車両の列は 2 種類の文字 I, O のみからなる文字列で表され，2 つの車庫の情報はそれぞれ長さ M の文字列 S および長さ N の文字列 T として与えられる．各文字が 1 つの車両を表し，その文字は車両の種類と同じである．文字列の 1 文字目は最も車庫の入り口に近い車両を表し，末尾の文字が車庫の最も奥にある車両を表す．

制限

$1 \leq M \leq 2000$ 文字列 S の長さ

$1 \leq N \leq 2000$ 文字列 T の長さ

入力

標準入力から以下のデータを読み込め．

- 1 行目には M, N が空白区切りで書かれている．
- 2 行目には文字列 S が書かれている．
- 3 行目には文字列 T が書かれている．

出力

標準出力に，編成できる列車の長さの最大値を表す整数を 1 行で出力せよ．列車が 1 つも編成できない場合は，0 を出力せよ．

採点基準

採点用データのうち，配点の 20%分については， $M \leq 10, N \leq 10$ を満たす．

採点用データのうち，配点の 50%分については， $M \leq 50, N \leq 50$ を満たす．



入出力例

入力例 1	出力例 1
5 5 OI00I 00IOI	7

S によって表される車庫を車庫 S とし, T によって表される車庫を車庫 T としよう. このとき, たとえば車庫 S から最初の 1 車両, 車庫 T から最初の 2 車両を出して待機させた後, 車庫 S , 車庫 S , 車庫 T , 車庫 S , 車庫 S , 車庫 T , 車庫 T の順番に車両を出せば, 長さ 7 の列車 $IOIOIOI$ を編成できる.

他にも, 車庫 S から最初の 1 車両, 車庫 T から最初の 2 車両を出して待機させた後, 車庫 T , 車庫 T , 車庫 S , 車庫 S , 車庫 T , 車庫 S , 車庫 S の順番に車両を出すことでも長さ 7 の列車を編成できる. これより長い列車を編成することはできないので 7 を出力する.

入力例 2	出力例 2
5 9 IIIII IIIIIIIII	1

1 つの車両のみからなる列車 I も列車としての条件を満たすことに注意せよ.



3

現代的な屋敷 (Modern Mansion)

あなたは、ある大きな屋敷に迷い込んでしまった。この屋敷は正方形の部屋が東西南北に格子状に、東西方向に M 列、南北方向に N 行の合計 $M \times N$ 個並んだ構造をしている。西から x 列目 ($1 \leq x \leq M$)、南から y 行目 ($1 \leq y \leq N$) にある部屋を (x, y) で表す。

東西南北に隣り合う2部屋の間は、壁の中央にある扉により結ばれている。それぞれの扉は、閉じていて通行不可能な状態か、開いていて通行可能な状態のいずれかにある。扉が開いているとき、それらの部屋の中央間を移動するのに1分間かかる。また、いくつかの部屋の中央にはスイッチがあり、スイッチを1分間押し続けると、屋敷内のすべての扉の開閉の状態が切り替わる。

今、東西に隣り合う2部屋を結ぶすべての扉は閉じていて、南北に隣り合う2部屋を結ぶすべての扉は開いている。あなたは今部屋 $(1, 1)$ の中央にいて、部屋 (M, N) の中央まで最短時間で移動したい。

課題

屋敷の大きさ M, N および、スイッチのある K 個の部屋の位置 $(X_1, Y_1), (X_2, Y_2), \dots, (X_K, Y_K)$ が与えられる。東西に隣り合う2部屋を結ぶすべての扉は閉じていて、南北に隣り合う2部屋を結ぶすべての扉は開いている状態から始めて、部屋 $(1, 1)$ の中央から部屋 (M, N) の中央まで移動するのに最短で何分かかかるかを求めるプログラムを作成せよ。ただし、部屋 (M, N) に辿り着くことができないときはそれを指摘せよ。

制限

$2 \leq M \leq 100\,000$	屋敷の東西方向の部屋の個数
$2 \leq N \leq 100\,000$	屋敷の南北方向の部屋の個数
$1 \leq K \leq 200\,000$	スイッチのある部屋の個数
$1 \leq X_i \leq M$	スイッチのある部屋の東西方向の位置
$1 \leq Y_i \leq N$	スイッチのある部屋の南北方向の位置

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 M, N, K が空白を区切りとして書かれている。 M は屋敷の東西方向の部屋の個数、 N は屋敷の南北方向の部屋の個数、 K はスイッチのある部屋の個数を表す。
- 続く K 行のうちの i 行目 ($1 \leq i \leq K$) には、整数 X_i, Y_i が空白を区切りとして書かれている。これは、部屋 (X_i, Y_i) の中央にスイッチがあることを表す。 K 個の組 $(X_1, Y_1), (X_2, Y_2), \dots, (X_K, Y_K)$ は互いに異なる。



出力

標準出力に、移動に最短で何分かかかるかを表す整数を 1 行で出力せよ。ただし、部屋 (M, N) に辿り着くことができないときは代わりに整数 -1 を出力せよ。

採点基準

採点用データのうち、配点の 20% 分については、 $M \leq 1000, N \leq 1000$ を満たす。

採点用データのうち、配点の 30% 分については、 $K \leq 2000$ を満たす。

採点用データのうち、配点の 50% 分については、これら 2 つの条件の少なくとも一方を満たす。また、これら 2 つの条件の両方を満たすような採点用データはない。

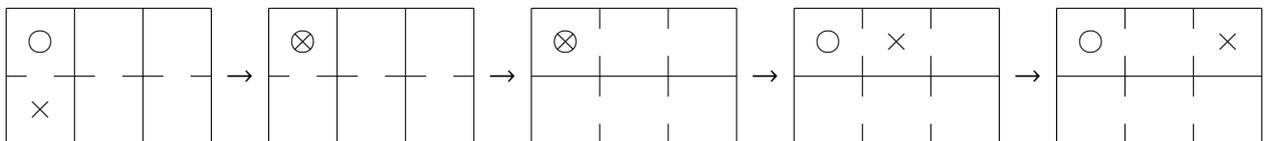
入出力例

入力例 1	出力例 1
3 2 1 1 2	4

この例では、以下の行動によって 4 分間で部屋 $(1, 1)$ の中央から部屋 $(3, 2)$ の中央へ移動することができ、これが最短である。

1. 部屋 $(1, 2)$ の中央へ移動する。
2. 部屋 $(1, 2)$ の中央のスイッチを押す。
3. 部屋 $(2, 2)$ の中央へ移動する。
4. 部屋 $(3, 2)$ の中央へ移動する。

このときの屋敷の様子が以下の図に表されている。図では右方向が東、上方向が北であり、 \times 印はあなたの位置、 \circ 印はスイッチを表す。



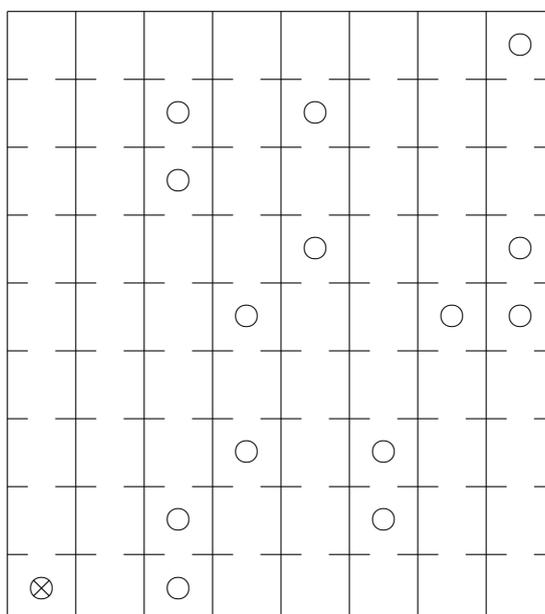
入力例 2	出力例 2
3 2 1 2 1	-1

この例では、あなたは部屋 $(3, 2)$ に辿り着くことができない。



入力例 3	出力例 3
8 9 15	25
3 1	
3 2	
3 7	
3 8	
1 1	
4 5	
4 3	
5 6	
5 8	
6 3	
6 2	
7 5	
8 9	
8 6	
8 5	

この例では、最初の屋敷の様子は以下の図のようになっている。部屋(1,1)や部屋(M,N)の中央にスイッチがある可能性もあることに注意せよ。



4

JOIOI の塔 (Tower of JOIOI)

JOIOI の塔とは、1 人で遊ぶ円盤を使ったゲームである。

このゲームは、J, O, I のいずれかの文字が書かれたいくつもの円盤を用いて行う。円盤は直径が互いに異なり、ゲーム開始時にはこれらの円盤は直径の大きいものから順に下から上に向かって積まれている。あなたは、これらの円盤を用いて、出来るだけ多くのミニ JOIOI の塔を作りたい。ミニ JOIOI の塔とは 3 枚の円盤からなり、円盤の直径が小さいものから順に読んで JOI もしくは IOI と読めるものである。ただし、同じ円盤を 2 度以上使うことはできない。

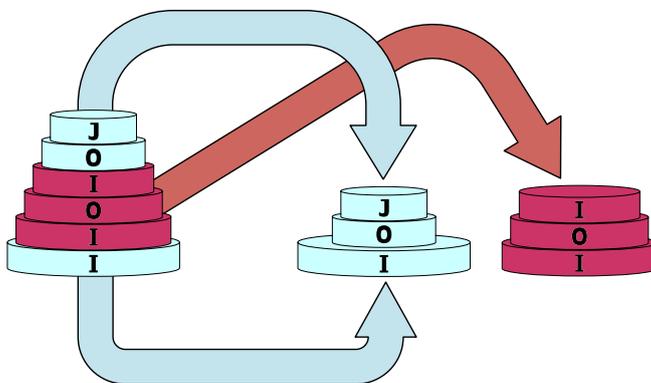


図: JOIOII からはミニ JOIOI の塔が 2 つ作れる

課題

用意された円盤に書かれた文字がそれぞれ円盤の直径が小さいものから順に長さ N の文字列 S として与えられる。これらの円盤を使って作ることのできるミニ JOIOI の塔の個数の最大値を求めるプログラムを作成せよ。

制限

$1 \leq N \leq 1000000$ 文字列 S の長さ

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。 N は文字列 S の長さを表す。
- 2 行目には文字列 S が書かれている。



出力

標準出力に、作ることのできるミニ JOIOI の塔の数の最大値を表す整数を1行で出力せよ。

採点基準

採点用データのうち、配点の10%分については、 $N \leq 15$ を満たす。

採点用データのうち、配点の30%分については、 $N \leq 50$ を満たす。

採点用データのうち、配点の50%分については、 $N \leq 3000$ を満たす。

入出力例

入力例 1	出力例 1
6 JOIIOI	2

JOIIOI は JOI および IOI をそれぞれ1つずつ部分列として含んでおり、作ることのできるミニ JOIOI の塔は2つである。

入力例 2	出力例 2
5 JOIOI	1

部分列として JOI および IOI を含んでいるが、文字を2度以上使うことはできないため同時に取り出すことはできない。

入力例 3	出力例 3
6 JOIOII	2

この入出力例は問題文中の例に対応している。

入力例 4	出力例 4
15 JJJOII00JOJI0II0	4



5

バブルソート (Bubble Sort)

バブルソートとは、列をソートするアルゴリズムの 1 つである。長さ N の数列 A を昇順にソートしたいとしよう。バブルソートは、隣り合う 2 つの数で大小関係が崩れているものがあれば、それらの位置を交換する。これを、数列を前から順に走査しながら行う。すなわち、 $A_i > A_{i+1}$ となっている場所があれば、その 2 数を交換するというのを、 $i = 1, 2, \dots, N-1$ に対してこの順で行うのが 1 回の走査である。この走査を $N-1$ 回繰り返すことで、数列を昇順にソートできることが知られている。

数列 A のバブルソートによる交換回数とは、数列 A に上記のアルゴリズムを適用した時に、整数の交換が行われる回数である。(バブルソートとして知られるアルゴリズム及び実装には、ループの順番や範囲、及び終了条件など、細かな差異がある場合がある。ただし、同じ数列に適用した際の整数の交換回数はそれらの差異により変化しないことが知られている。)

例えば、以下のプログラムは長さ n の整数の配列 a をバブルソートによりソートする関数を C 言語で記述したものである。

```
void bubble_sort(int *a, int n) {
    int i, j;
    for (i = 0; i < n - 1; ++i) {
        for (j = 0; j < n - 1; ++j) {
            if (a[j] > a[j + 1]) {
                /* 以下 3 行が 1 回の整数の交換に相当 */
                int x = a[j];
                a[j] = a[j + 1];
                a[j + 1] = x;
            }
        }
    }
}
```

課題

長さ N の数列 A が与えられる。数列 A の任意の場所の 2 つの整数を 1 回だけ交換した数列 A' を作る。数列 A' のバブルソートによる交換回数の最小値を求めるプログラムを作成せよ。(最初に交換する 2 つの整数は必ずしも隣り合っている必要はないことに注意せよ。)

制限

- $1 \leq N \leq 100\,000$ 数列 A の長さ
- $1 \leq A_i \leq 1\,000\,000\,000$ 数列 A に含まれる数字の大きさ



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。 N は数列 A の長さを表す。
- 続く N 行のうちの i 行目 ($1 \leq i \leq N$) には、整数 A_i が書かれている。これは数列 A の i 番目の整数を表す。

出力

標準出力に、数列 A' のバブルソートによる交換回数の最小値を表す整数を 1 行で出力せよ。

採点基準

採点用データのうち、配点の 10%分については、 $N \leq 1000$ を満たし、任意の $i, j (1 \leq i < j \leq N)$ について $A_i \neq A_j$ を満たす。

採点用データのうち、配点の 30%分については、 $N \leq 5000$ を満たし、任意の $i, j (1 \leq i < j \leq N)$ について $A_i \neq A_j$ を満たす。

採点用データのうち、配点の 80%分については、任意の $i, j (1 \leq i < j \leq N)$ について $A_i \neq A_j$ を満たす。

入出力例

入力例 1	出力例 1
5	0
10	
3	
6	
8	
1	

数列 A の最初の 10 と最後の 1 を交換することになると、数列 A' はソート済みの列となり、バブルソートの交換回数は 0 となる。

入力例 2	出力例 2
5	2
3	
1	
7	
9	
5	



数列 A の 3 番目の 7 と最後の 5 を交換することで、数列 A' は 3,1,5,9,7 となる。 A' のバブルソートによる交換回数は 2 である。

入力例 3	出力例 3
3	1
1	
2	
3	

最初から数列 A がソートされている場合でも、数列 A' を作る際に交換を行わなければならない。