

第7回 日本情報オリンピック本選問題¹

2008年2月10日

受験生への注意事項

- 試験開始の指示のあるまで、問題が入った封筒を開けないで下さい。
- 問題は5問、試験時間は4時間（13時～17時）です。
- 提出方法や注意事項は overview sheet に記述されています。
- まず、cygwin の home directory
/home/[受験番号] (C:¥cygwin¥home¥[受験番号])
にある joi2008.exe を実行してください。パスワードは
J=xDaw3B
です。
- 前項の処理を行うと home directory に joi2008 という directory が作成され、
その下に 2008-ho-t1, 2008-ho-t2, 2008-ho-t3, 2008-ho-t4, 2008-ho-t5 の
5つの directory が作成されます。解答はこれらの directory で、overview sheet
の指示に従い行ってください。また、これらの directory には、各問の
 - 入力データ例とその入力に対する正解出力データ
(どの入力データ例についても、対応する正解出力データがあります。)
 - 採点時と同じ方法でコンパイルするスクリプト
 - 上のスクリプトでコンパイルしたプログラムを、指定した入力データ例
に対して実行するスクリプト
 - 上のスクリプトでコンパイルしたプログラムを、指定した入力データ例
に対して実行しその出力を正解と比較するスクリプトがあります。

情報オリンピック日本委員会

¹Copyright ©2008 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

碁石ならべ

問題

白と黒の碁石をテーブルの上にならべて遊ぶ。まずテーブルの左端に碁石を置く。次に左から 2 番目の場所に碁石を置く。これを n 回繰り返して n 個の碁石を横一列にならべる。ただし、新しく i 番目の碁石を置く際には、次のルールに従ってテーブル上の碁石を置き換える。

- i が奇数の場合: テーブルに置いてあった碁石は置き換えず、新しい碁石を左から i 番目に置く。
- i が偶数の場合: 新しく左から i 番目に置く碁石の色とテーブル上の右端の碁石の色が同じ場合は、テーブル上の碁石は置き換えず、新しい碁石を左から i 番目に置く。そうでない場合、すなわち、新しく左から i 番目に置く碁石の色とテーブル上の右端の碁石の色が異なる場合は、まずテーブル上の右端の連続した同色の碁石を全て取り除き、 i 番目の碁石と同色の碁石に置き換える。そしてテーブルの右端に i 番目の碁石を置く。

例えば、最初の 7 個の碁石を置いた時点で、

となっていたとする (\square は白の碁石を、 \blacksquare は黒の碁石を表す)

- 8 番目の碁石が白 (\square) の場合は、右端の碁石と同色なので、そのまま置く。したがって、テーブル上の碁石は

となる。

- 8 番目の碁石が黒 (\blacksquare) の場合は、右端の碁石 (\square) と色が異なるので、まずテーブルの右端にある 3 個の連続した白い碁石 (\square) を取り除き、黒い碁石 (\blacksquare) に置き換える。そして右端に 8 番目の碁石を置く。したがって、テーブル上の碁石は

となる。

入力として置く碁石の順番が与えられたとき、 n 個の碁石をならべ終わった後、テーブル上に置いてある白い碁石の個数を求めるプログラムを作成せよ。

入力

入力ファイルのファイル名は `input.txt` である。

1 行目には正整数 n ($1 \leq n \leq 100000$) が書かれている。2 行目以降の第 $i+1$ 行目 ($1 \leq i \leq n$) には、 i 番目に置く碁石の色を表す整数 c_i が書かれており、 c_i が 0 なら i 番目に置く碁石の色が白であることを、1 なら i 番目に置く碁石の色が黒であることを表す。

採点用データのうち、配点の 50% 分については、 $n \leq 10000$ を満たす。

出力

出力ファイルのファイル名は `output.txt` である。

`output.txt` は、 n 個の碁石をならべ終わった後にテーブル上に置いてある白い碁石の個数だけを含む 1 行からなる。

入出力の例

例 1

`input.txt`

8
1
0
1
1
0
0
0
0
0

`output.txt`

6

例 2

`input.txt`

8
1
0
1
1
0
0
0
0
1

`output.txt`

2

2

共通部分文字列

問題

2 個の文字列が与えられたとき、両方の文字列に含まれる文字列のうち最も長いものを探し、その長さを答えるプログラムを作成せよ。

ここで、文字列 s が文字列 t に含まれるとは、 t の中に s が連続して現れることをいう。空文字列、すなわち長さ 0 の文字列は、どんな文字列にも含まれる。例えば、文字列 ABRACADABRA には次の文字列が含まれる: ABRA, RAC, D, ACADABRA, ABRACADABRA, 空文字列など。一方、文字列 ABRACADABRA には次の文字列は含まれない: ABRC, RAA, BA, K など。

例

例 1: 文字列として ABRACADABRA と ECADADABRBCRDARA が与えられた場合、両方に含まれる文字列には CA や CADA や ADABR や空文字列などがある。そのうち最も長いのは ADABR であり、その長さは 5 である。2 個の文字列の中に含まれる ADABR の位置を図 2-1 に示す。

```
ABRACADABRA
ECADADABRBCRDARA
```

図 2-1 例 1 の文字列の中に現れる ADABR の位置

例 2: 文字列として UPWJCIRUCAXIIRGL と SBQNYBSBZDFNEV が与えられた場合、両方に含まれる文字列は空文字列のみであり、その長さは 0 である。

入力

入力ファイルのファイル名は input.txt である。

入力は 2 行からなり、1 行目に 1 個目の文字列が、2 行目に 2 個目の文字列が与えられる。文字列は英大文字からなり、各々の文字列の長さは 1 以上 4000 以下である。

採点用データのうち、配点の 30% 分については、各々の文字列の長さは 1 以上 50 以下である。

出力

出力ファイルのファイル名は `output.txt` である .

`output.txt` は , 与えられた 2 個の文字列の両方に含まれる文字列のうち最も長いものの長さだけを含む 1 行からなる .

入出力の例

例に対応する入出力は以下の通りである .

例 1

`input.txt`

```
ABRACADABRA
ECADADABRBCRDARA
```

`output.txt`

5

例 2

`input.txt`

```
UPWJCIRUCAXIIRGL
SBQNYBSBZDFNEV
```

`output.txt`

0

3

ダーツ

問題

あなたは以下のルールでダーツゲームをすることになった。

あなたは、矢を的(まと)に向かって4本まで投げることができる。必ずしも4本全てを投げる必要はなく、1本も投げなくてもかまわない。的は N 個の部分に区切られていて、各々の部分に点数 P_1, \dots, P_N が書かれている。矢が刺さった場所の点数の合計 S があなたの得点の基礎となる。 S があらかじめ決められたある点数 M 以下の場合には S がそのままあなたの得点となる。しかし、 S が M を超えた場合はあなたの得点は0点となる。

的に書かれている点数と M の値が与えられたとき、あなたが得ることのできる点数の最大値を求めるプログラムを作成せよ。

入力

入力ファイルのファイル名はinput.txtである。

1行目には、整数 $N(1 \leq N \leq 1000)$ と $M(1 \leq M \leq 200000000 = 2 \times 10^8)$ がこの順に空白区切りで書かれている。2行目以降の第 $1+i$ 行目($1 \leq i \leq N$)には、 $P_i(1 \leq P_i \leq 100000000 = 10^8)$ が書かれている。

採点用データのうち、配点の20%分は $N \leq 100$ を満たし、配点の50%分は $N \leq 300$ を満たす。

出力

出力ファイルのファイル名はoutput.txtである。

output.txtは1行だけからなり、あなたが得ることのできる点数の最大値を含む。

入出力の例

例 1

input.txt

```
4 50
3
14
15
9
```

output.txt

```
48
```

この例では、15 点の部分に 3 本、3 点の部分に 1 本の矢が刺さった場合にあなたの得点は最大になり、その得点は 48 点である。

例 2

input.txt

```
3 21
16
11
2
```

output.txt

```
20
```

この例では、16 点の場所に 1 本、2 点の場所に 2 本の矢が刺さった場合にあなたの得点は最大になり、その得点は 20 点である。

4

ぴよんぴよん川渡り

問題

ある川で、一方の岸から石の上を次々と飛び移って反対側の岸まで行くという少々危険なゲームがはやっている。

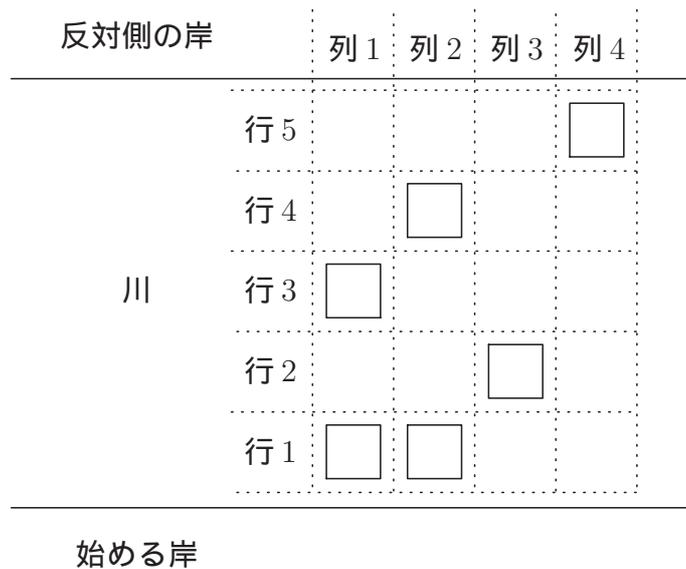


図 4-1 石の位置の例

今、図 4-1 のように、石はマス目の上に存在すると考える。行の数は n である。図 4-1 では $n = 5$ である。

このゲームでは、一方の岸から始めて、通常のジャンプまたは m 回以下の一行飛ばしのジャンプをして反対側の岸まで渡る。通常のジャンプとは、今いる行の一つ先の行の岸または石のどれかに飛び移ることであり、一行飛ばしのジャンプとは、今いる行の二つ先の行の岸または石のどれかに飛び移ることである。始める岸の一つ先の行は 1 行目、二つ先の行は 2 行目であり、 $n - 1$ 行目の二つ先の行、及び n 行目の一つ先の行は反対側の岸であるとする。

さて、このゲームをできるだけ安全に成し遂げるために、ジャンプの危険度というものを考えることにした。それぞれの石には、滑りやすさが定まっている。石から石へ飛び移る際のジャンプの危険度は、通常のジャンプであっても、一行飛ばしのジャンプであっても、

$$(\text{今いる石の滑りやすさ} + \text{飛び移る先の石の滑りやすさ}) \times (\text{横の移動距離})$$

で定める。ただし、横の移動距離とは、列の番号の差である。また、岸から石へ、あるいは石から岸へ飛び移るジャンプの危険度は0とする。

入力として n, m 、それぞれの石の位置と滑りやすさが与えられたとき、反対側の岸まで到達する際のジャンプの危険度の合計の最小値を求めるプログラムを作成せよ。与えられる入力データは必ず反対側の岸まで到達することができるようになっており、同じマス目に石は2個以上存在しない。

入力

入力ファイルのファイル名は `input.txt` である。

入力ファイルの1行目には、空白を区切りとして2個の整数 n, m が書かれている。これは、行の数と、一行飛ばしのジャンプの許される回数を表す。 n, m はそれぞれ $2 \leq n \leq 150, 0 \leq m \leq \frac{(n+1)}{2}$ を満たす。

続く n 行には、それぞれの行にある石の情報が書かれている。 $i+1$ 行目 ($1 \leq i \leq n$) には、1つの整数 k_i ($0 \leq k_i \leq 10$) と、それに続き $2 \times k_i$ 個の整数が空白で区切られ書かれている。これらは、始める岸から数えて i 番目の行にある石の情報を表す。

k_i はその行に存在する石の個数を表し、それに続く $2 \times k_i$ 個の整数については、 $2 \times j - 1$ 個目 ($1 \leq j \leq k_i$) の整数 $x_{i,j}$ はその行の j 個目の石の列の番号を、 $2 \times j$ 個目の整数 $d_{i,j}$ はその行の j 個目の石の滑りやすさをそれぞれ表す。 $x_{i,j}, d_{i,j}$ はそれぞれ $1 \leq x_{i,j}, d_{i,j} \leq 1000$ を満たす。

採点用データのうち、配点の20%分については、 $n \leq 6$ を満たし、配点の別の20%分については、 $m = 0$ を満たす。

出力

出力ファイルのファイル名は `output.txt` である。

`output.txt` は、反対側の岸まで到達する際のジャンプの危険度の合計の最小値を表す1つの整数を含む1行からなる。

例

図4-2において、石に書かれた数字はそれぞれの石の滑りやすさを表すとする。矢印で示された順番で石を渡るとき、それぞれのジャンプの危険度は、順番に0, $(2+2) \times 1 = 4$, $(2+1) \times 1 = 3$, $(1+4) \times 2 = 10$, 0であり、合計は17となる。このとき、ジャンプの危険度の合計は最小となる。

この例は入出力例に対応している。

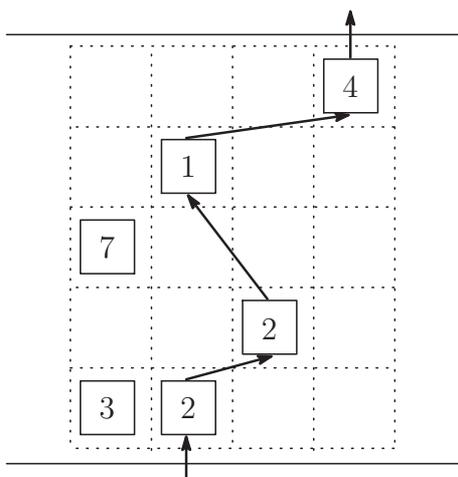


図 4-2 経路の例

入出力の例

上の例に対応する入出力は例 1 の通りである。

例 1

input.txt

```
5 1
2 1 3 2 2
1 3 2
1 1 7
1 2 1
1 4 4
```

output.txt

17

例 2

input.txt

```
5 0
2 1 3 2 2
1 3 2
1 1 7
1 2 1
1 4 4
```

output.txt

40

5

ペンキの色

問題

情報オリンピックの宣伝のために、長方形のベニヤ板にペンキを塗り看板を制作したい。ベニヤ板には色を塗りたくないところにあらかじめ何枚かの長方形のマスキングテープが貼られている。そこでマスキングテープで区切られた領域ごとに別々の色を使いペンキを塗ることにした。例えば、図 5-1 の場合は 5 色のペンキを使う。

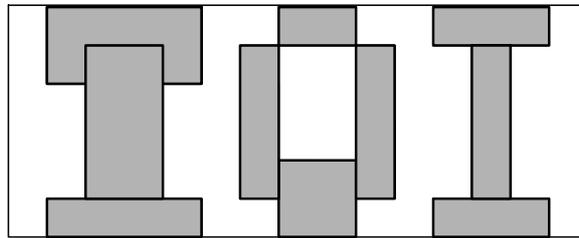


図 5-1 看板の例

入力としてマスキングテープを貼る位置が与えられた時、使うペンキの色の数を求めるプログラムを作成せよ。ただし、ベニヤ板全体がマスキングテープで覆われることはなく、全てのマスキングテープの辺はベニヤ板のいずれかの辺に平行である。

入力

入力ファイルのファイル名は `input.txt` である。

1 行目にはベニヤ板の幅 w ($1 \leq w \leq 1000000$ となる整数) と高さ h ($1 \leq h \leq 1000000$ となる整数) がこの順に空白区切りで書かれている。

2 行目にはマスキングテープの数 n ($1 \leq n \leq 1000$ となる整数) が書かれている。続く 3 行目以降の $2+i$ 行目 ($1 \leq i \leq n$) には、 i 番目に貼るマスキングテープの左下の座標 (x_1, y_1) と、右上の座標 (x_2, y_2) が x_1, y_1, x_2, y_2 ($0 \leq x_1 < x_2 \leq w, 0 \leq y_1 < y_2 \leq h$ となる整数) の順に空白区切りで書かれている。

ただし、ベニヤ板の左下の角の座標は $(0, 0)$ で右上の角の座標は (w, h) である。

採点用データのうち、配点の 30% 分は $w \leq 100, h \leq 100, n \leq 100$ を満たす。

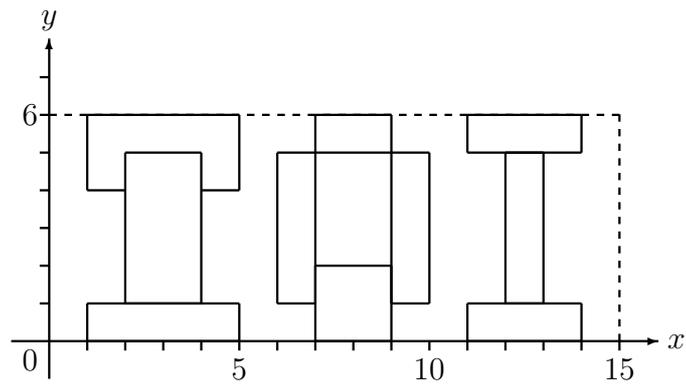


図 5-2 図 5-1 の看板の入力例

出力

出力ファイルのファイル名は `output.txt` である。
`output.txt` は 1 行だけからなり、その 1 行は使うペンキの色数だけを含む。

入出力の例

次の例は図 5-1 の場合である。

`input.txt`

```

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

```

`output.txt`

```

5

```