13th JOI Selection Final Selection Tasks (2014)

Authors: JCIOI (Japanese Committee for the IOI)

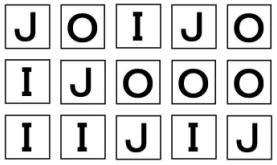
Translators: Jasmine Chua, France-ioi

License: Creative Commons Attributions-ShareAlike 4.0 International License (CC BY-SA 4.0)

**1. JOI Emblem**

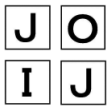
The committee members of the Informatics Olympiad are designing a new JOI flag to support the athletes participating in the competition in Taiwan.

The JOI flag is a rectangle, which can be divided into a grid with M vertical columns and N horizontal rows. Each square in the grid is written with 1 of the following 3 letters: J, O, or I.



Example of a JOI flag

The committee members are also designing a separate JOI emblem. The JOI emblem is a square, which can be divided into a grid with 2 vertical columns and 2 horizontal rows. Each square in the grid is written with 1 of the following 3 letters: J, O, or I.



Example of a JOI emblem

The design of the JOI emblem, which measures 2 by 2, may appear on the design of the JOI flag (without rotation or flipping). Any replication of the JOI emblem on the JOI flag is counted, even if it overlaps with another replication.

The committee members have an old JOI flag and 1 piece of blank paper. The blank paper is the size of one of the squares on the JOI flag, and 1 of the following 3 letters: J, O, or I, can be written on it. The committee members will choose one of the methods below to create the new JOI flag.

* They will not change the old JOI flag in any way and use it as the new JOI flag. The blank paper is not used.
* They will write 1 letter on the blank paper and paste it over 1 square of the old JOI flag to change its letter. This will become the new JOI flag.

The committee members hope to have the JOI emblem design appear as many times as possible on the new JOI flag. You have been tasked to find out the maximum number of times the JOI emblem design will appear on the new JOI flag.

**QUESTION**

Given the old JOI flag and the JOI emblem design, create a program to find out the maximum number of times the JOI emblem design will appear on the new JOI flag.

**INPUT**

In the first row, 2 integers are written with a space between them.

* M, representing the number of rows on the JOI flag.
* N, representing the number of columns on the JOI flag.

For the next M rows, a string of N characters is written.

* Each character can be one of the following 3 letters: J, O, or I. The jth character in the ith row, where 1 ≤ i ≤ M and 1 ≤ j ≤ N, represents the letter written on the old JOI flag in the square that is ith from the top and jth from the left.

For the following 2 rows, 2 characters are written.

* Each character can be one of the following 3 letters: J, O, or I. The jth character in the ith row, where 1 ≤ i ≤ 2 and 1 ≤ j ≤ 2, represents the letter on the JOI emblem in the square that is ith from the top and jth from the left.

**OUTPUT**

Output one row, with the maximum number of times the JOI emblem design will appear on the new JOI flag.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 2 ≤ M ≤ 1 000
* 2 ≤ N ≤ 1 000

**SUBTASKS**

**Subtask 1 [30 points]**

Fulfil the following constraints:

* M ≤ 50
* N ≤ 50

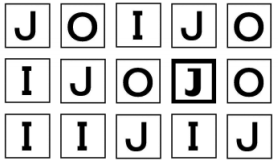
**Subtask 2 [70 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

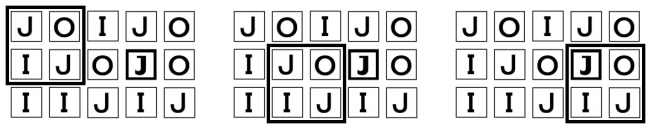
|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 3 5  JOIJO  IJOOO  IIJIJ  JO  IJ | 3 |

The old JOI flag and JOI emblem design are as per the examples given in the question. If the blank paper is used to change the letter that is 2nd from the top and 4th from the left to a J, the new JOI flag will appear like this:



Example of changing 1 letter on the JOI flag

The JOI emblem design appears in 3 different locations on this new JOI flag.



Locations where the JOI emblem design appears

Since 3 is the maximum number of times the JOI emblem design appears on the new JOI flag, the output is 3.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 2 6  JOJOJO  OJOJOJ  OJ  JO | 2 |

Note that the same maximum output is obtained with or without using the blank paper.

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| 2 2  JI  IJ  JJ  JJ | 0 |

In this input example, the JOI emblem design will not appear on the JOI flag, no matter what is changed.

**2. IOI Buns**

Incredible Okashi Inc. is a company that produces incredibly delicious pastries. Its name is abbreviated into IOI. IOI Company has decided to produce and sell IOI buns. They made M types of buns, with 1 bun per type. All the buns are of the same size, but contain different fillings, so are priced differently. Bun i, where 1 ≤ i ≤ M, costs Pi yen.

By the way, do you know of the company called Just Odd Inventions? They are a company that makes odd inventions, and its name is abbreviated into JOI. IOI Company has decided to order some high-quality boxes from JOI Company to sell their buns in. JOI Company makes N types of boxes, and Box j, where 1 ≤ j ≤ N, which costs Ej yen, can hold a maximum of Cj buns. IOI Company ordered some types of boxes (between 0 and N), divided their buns into the boxes and sold the buns as sets. The price of each set is determined by the total price of the buns within.

What is the maximum profit IOI Company makes after selling all their bun sets? Profit is calculated by taking away the total price of the boxes from the total price of the buns sets sold. Buns that were not put into boxes were eaten by the IOI Company employees and do not count towards profit.

**QUESTION**

Given the price of each bun, and the size and price of each box, create a program to find out the maximum profit IOI Company makes after selling all their bun sets.

**INPUT**

In the first row, 2 integers are written with a space between them.

* M, representing the number of types of buns.
* N, representing the number of types of boxes.

For the next M rows, in the ith row, where 1 ≤ i ≤ M, one integer is written.

* Pi, representing the price of Bun i.

For the following N rows, in the jth row, where 1 ≤ j ≤ N, two integers are written with a space between them.

* Cj, representing the number of buns Box i can hold.
* Ej, representing the price of Box i.

**OUTPUT**

Output one row, with the maximum profit IOI Company makes after selling all their bun sets.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 1 ≤ M ≤ 10 000
* 1 ≤ N ≤ 500
* 1 ≤ Pi ≤ 10 000, where 1 ≤ i ≤ M
* 1 ≤ Cj ≤ 10 000, where 1 ≤ j ≤ N
* 1 ≤ Ej ≤ 10 000, where 1 ≤ j ≤ N

**SUBTASKS**

**Subtask 1 [25 points]**

Fulfil the following constraint:

* N ≤ 10

**Subtask 2 [35 points]**

Fulfil the following constraint:

* Cj ≤ 10, where 1 ≤ j ≤ N

**Subtask 3 [40 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 4 3  180  160  170  190  2 100  3 120  4 250 | 480 |

In this input example, IOI Company will order Box 1 (100 yen) and Box 2 (120 yen). Bun 1 and Bun 2 will be placed into Box 1 and sold as a set for 180 + 160 = 340 yen. Bun 3 and Bun 4 will be placed into Box 2 and sold as a set for 170 + 190 = 360 yen. The profit will be 700 − 220 = 480 yen.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 2 2  1000  2000  1 6666  1 7777 | 0 |

In this input example, IOI Company should not buy any boxes to maximize their profit.

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| 10 4  200  250  300  300  350  400  500  300  250  200  3 1400  2 500  2 600  1 900 | 450 |

**3. Baumkuchen**

JOI is eating desserts with his sisters JOIKO and JOIMI. Today, they are eating their favorite dessert, baumkuchen.

Baumkuchen is a tube-shaped pastry, as shown in the diagram below. To share between the 3 siblings, JOI needs to cut the pastry 3 times to split it into 3 pieces. However, this baumkuchen is as hard as a log, so it is not easy to cut. The baumkuchen has been precut at N locations, so JOI chooses from these precuts to make his actual cuts. The precuts are numbered from 1 to N in a clockwise direction. The piece between Precut i and Precut (i + i), where 1 ≤ i ≤ N − 1, has a size of Ai. The piece between Precut N and Precut 1 has a size of AN.

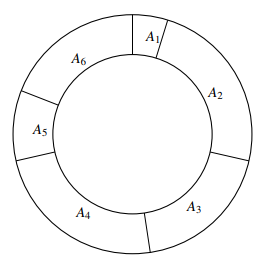


Diagram 1: Example of the baumkuchen where N = 6. A1 = 1, A2 = 5, A3 = 4, A4 = 5, A5 = 2, A6 = 4

JOI loves his sisters, so after cutting up the baumkuchen, he will choose the smallest piece and give the 2 remaining pieces to each sister. However, JOI also loves baumkuchen, so he would like to eat as much as possible. What is the size of the piece JOI gets, if he cuts in a way to maximize his piece, even as he gets the smallest piece of the 3?

**QUESTION**

Given the number of precuts N and the size of each piece A1, ..., AN, create a program to find out the maximum size of the smallest piece if the baumkuchen is cut into 3 pieces.

**INPUT**

In the first row, 1 integer is written.

* N, representing the number of precuts on the baumkuchen.

For the next N rows, in the ith row, where 1 ≤ i ≤ N, 1 integer is written.

* Ai, representing the size of the piece between Precut i and Precut (i + 1). If i = N, then it represents the size of the piece between Precut N and Precut 1.

**OUTPUT**

Output one row, with the maximum size of the smallest piece if the baumkuchen is cut into 3 pieces.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 3 ≤ N ≤ 100 000
* 1 ≤ Ai ≤ 1 000 000 000, where 1 ≤ i ≤ N

**SUBTASKS**

**Subtask 1 [5 points]**

Fulfil the following constraint:

* N ≤ 100

**Subtask 2 [15 points]**

Fulfil the following constraint:

* N ≤ 400

**Subtask 3 [30 points]**

Fulfil the following constraint:

* N ≤ 8 000

**Subtask 4 [50 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 6  1  5  4  5  2  4 | 6 |

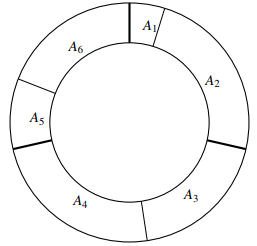


Diagram 2: JOI should make his cuts at Precuts 1, 3, and 5.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 30  1  34  44  13  30  1  9  3  7  7  20  12  2  44  6  9  44  31  17  20  33  18  48  23  19  31  24  50  43  15 | 213 |

**4. Sugar Glider**

There are N eucalyptus trees in the forest where Sugar Glider JOI lives, numbered from 1 to N. Tree i has a height of Hi.

There are M pairs of trees which JOI can directly glide between, and JOI takes a set amount of time to glide between each pair. When JOI is gliding from tree to tree, he will fall towards the ground at a rate of 1 meter per second. In other words, if JOI is at a height h meters from the ground, and takes t seconds to glide between trees, he will land at a height (h − t) meters from the ground. JOI cannot glide from tree to tree if (h − t) is less than 0, or if (h − t) is taller than the destination tree’s height.

Furthermore, JOI can also move vertically up and down a tree, from a height 0 meters from the ground to the tree's maximum height. JOI takes 1 second to move 1 meter vertically up or down a tree.

JOI wants to move from a position X meters from the ground on Tree 1, to the top of Tree N (a position HN meters from the ground). He wants to know the minimum time needed to move there.

**QUESTION**

Given the height of each tree, pairs of trees that JOI can glide between, and the height of JOI's starting position, create a program to find out the minimum time needed to move to the top of Tree N.

**INPUT**

In the first row, 3 integers are written with a space between them.

* N, representing the number of trees.
* M, representing the number of pairs of trees JOI can glide between.
* X, representing JOI's initial height on Tree 1 in meters.

For the next N rows, in the ith row, where 1 ≤ i ≤ N, one integer is written.

* Hi, representing the height of Tree i.

For the next M rows, in the jth row, where 1 ≤ j ≤ M, three integers are written with a space between them.

* Aj and Bj, where 1 ≤ Aj ≤ N, 1 ≤ Bj ≤ N, and Aj ≠ Bj.
* Tj, representing the number of seconds it takes for JOI to glide between Tree Aj and Tree Bj.

Also, if 1 ≤ j < k ≤ M, then (Aj,Bj) ≠ (Ak, Bk) and (Aj,Bj) ≠ (Bk,Ak).

**OUTPUT**

Output one row, with the minimum time needed to move from a position X meters from the ground on Tree 1, to the top of Tree N. However, if this is not possible, output −1.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 2 ≤ N ≤ 100 000
* 1 ≤ M ≤ 300 000
* 1 ≤ Hi ≤ 1 000 000 000, where 1 ≤ i ≤ N
* 1 ≤ Tj ≤ 1 000 000 000, where 1 ≤ j ≤ M
* 0 ≤ X ≤ H1

**SUBTASKS**

**Subtask 1 [25 points]**

Fulfil the following constraints:

* N ≤ 1 000
* M ≤ 3 000
* Hi ≤ 100, where 1 ≤ i ≤ N
* Tj ≤ 100, where 1 ≤ j ≤ M

**Subtask 2 [25 points]**

Fulfil the following constraint:

* X = 0

**Subtask 3 [50 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 5 5 0  50  100  25  30  10  1 2 10  2 5 50  2 4 20  4 3 1  5 4 20 | 110 |

For example, JOI can move in such a manner:

1. Climb 50 meters on Tree 1.
2. Glide from Tree 1 to Tree 2.
3. Glide from Tree 2 to Tree 4.
4. Glide from Tree 4 to Tree 5.
5. Climb 10 meters on Tree 5.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 2 1 0  1  1  1 2 100 | -1 |

JOI cannot glide from Tree 1 to Tree 2.

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| 4 3 30  50  10  20  50  1 2 10  2 3 10  3 4 10 | 100 |

**5. Cutting Lines**

JOI is interested in paper craft. Today, he is making another paper craft.

First, JOI printed N cutting lines on one rectangular piece of paper, according to a design. Each cutting line is parallel to the paper's horizontal or vertical edge.

All parts that are cut out from the paper will be used in some way to create the final work. Of course, creating a work with multiple parts is difficult. JOI wants to know how many pieces of paper he will get after cutting the paper according to all the cutting lines.

**QUESTION**

Given the size of the paper and information about the N cutting lines, create a program to find out the number of pieces that will be obtained after cutting the paper according to all the cutting lines.

**INPUT**

In the first row, 3 integers are written with a space between them.

* W, representing the horizontal length of the paper.
* H, representing the vertical width of the paper.
* N, representing the number of cutting lines.

The 4 corners of the paper are represented as (0, 0), (W, 0), (0, H), and (W, H).

For the next N rows, in the ith row, where 1 ≤ i ≤ N, 4 integers are written with a space between them.

* Ai, Bi, Ci, and Di, where 0 ≤ Ai ≤ Ci ≤ W and 0 ≤ Bi ≤ Di ≤ H. This represents that Cutting Line i stretches between (Ai,Bi) and (Ci,Di). This line will be parallel to any of the paper's edges.

Also, a cutting line does not share any points with any cutting lines parallel to it, nor does it share any points with any edges parallel to it.

**OUTPUT**

Output one row, with the number of pieces that will be obtained.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 1 ≤ W ≤ 1 000 000 000
* 1 ≤ H ≤ 1 000 000 000
* 1 ≤ N ≤ 100 000

**SUBTASKS**

**Subtask 1 [5 points]**

Fulfil the following constraints:

* W ≤ 1 000
* H ≤ 1 000
* N ≤ 1 000

**Subtask 2 [5 points]**

Fulfil the following constraint:

* N ≤ 1 000

**Subtask 3 [20 points]**

Fulfil the following constraint:

* The number of pairs of cutting lines that share points is less than or equal to 100 000.

**Subtask 4 [20 points]**

Fulfil the following constraint:

* There is at least 1 cutting line leading from any point along a specific cutting line to any edge of the paper.

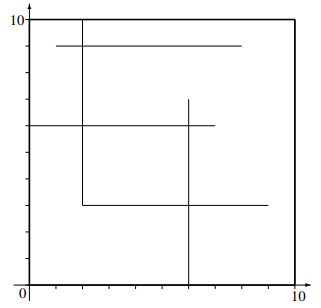
**Subtask 5 [50 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 10 10 5  6 0 6 7  0 6 7 6  2 3 9 3  2 3 2 10  1 9 8 9 | 4 |

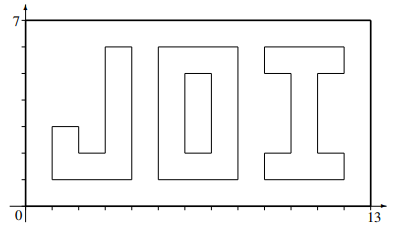
In this input example, the cutting lines are as shown:



According to these lines, the paper can be cut into 4 pieces. This fulfils the constraints of Subtask 4.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 13 7 28  1 1 4 1  1 1 1 3  2 2 3 2  2 2 2 3  1 3 2 3  3 2 3 6  4 1 4 6  3 6 4 6  5 1 8 1  5 1 5 6  6 2 7 2  6 2 6 5  7 2 7 5  6 5 7 5  8 1 8 6  5 6 8 6  9 1 12 1  9 1 9 2  9 2 10 2  12 1 12 2  11 2 12 2  10 2 10 5  9 5 10 5  9 5 9 6  11 2 11 5  11 5 12 5  12 5 12 6  9 6 12 6 | 5 |

In this input example, the cutting lines are as shown:



According to these lines, the paper can be cut into 5 pieces. This does not fulfil the constraints of Subtask 4.