11th JOI Selection Final Selection Tasks (2012)

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. JJOOII**

You are practicing programming in preparation for JOI (Japanese Olympiad in Informatics) and noticed that the preliminary questions for this year are questions on numbers, with no questions on character strings. You decided to secretly practice some questions on character strings to give yourself an advantage over your rivals.

Looking at past JOI questions, you noticed that many questions are based on character strings made up of 3 types of characters: J, O, and I. You decided to practice questions on partial character strings, such as "how many times does JOI appear in a given string". However, that is too easy, so you created the following question to make it more difficult.

Character String T is a partial character string of Character String S. By adding characters to the front and back of t (0 is fine), you can make s. For example, JJOOII is a partial character string of OJJOOIIOJOI. On the other hand, JOI is not a partial character string of JOOI.

Also, a JOI string of level k, where k > 0, is a character string made up of k 'J', k 'O', and k 'I', in that order. For example, JJOOII is a JOI string of level 2.

You want a character string that contains a JOI string of maximum level.

**QUESTION**

Given information about Character String S of length N, made up of 3 types of characters: J, O, and I, create a program to find out the maximum value of k, where a JOI string of level k is a partial character string of Character String S.

**CONSTRAINTS**

Length of S: 1 ≤ N ≤ 1000000 (= 106)

**INPUT**

In the first row, Character String S, made up of 3 types of characters: J, O, and I, is written.

**OUTPUT**

Output the maximum value of k, where a JOI string of level k is a partial character string of Character String S, in one row.

**SCORING CRITERIA**

For 20% of the allocated points, fulfil N ≤ 100.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| OJJOOIIOJOI | 2 |

OJJOOIIOJOI contains JJOOII, a JOI string of level 2. It does not contain a JOI string of level 3 or higher.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| IJJIIJJJ | 0 |

A JOI string of level 0 contains no characters.

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| JOIJOIJOIJOIJOI | 1 |

|  |  |
| --- | --- |
| INPUT 4 | OUTPUT 4 |
| OOJJJJJJJOOOOIIIII | 4 |

**2. Fun Card Game**

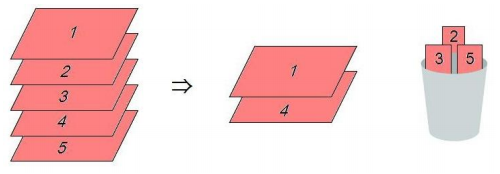
There are many cards, each with one integer, ranging from 1 to 1000, written on them. Anna and Bruno are using these cards to play the following game.

Both Anna and Bruno have a stack of cards: Anna's has A cards while Bruno's has B cards. From her stack, Anna chooses some cards (0 is fine) from anywhere in the stack and discards them, leaving the remaining cards as her new stack. From his stack, Bruno chooses some cards (0 is fine) from the top of the stack and some cards (0 is fine) from the bottom of the stack, and discards them, leaving the remaining cards as his new stack. After discarding the cards, they cannot change the order of the cards in their stacks. When the stacks they create become identical, they will get points corresponding to the number of cards in one stack. Identical stacks are when both stacks have the same number of cards, and the integer on the ith card from the top, where 1 ≤ i ≤ n, is identical in both stacks.

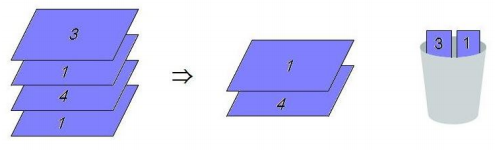
For example, Anna has a stack of 5 cards, and the integers written on them are 1, 2, 3, 4, and 5 from the top. Bruno has a stack of 4 cards, and the integers written on them are 3, 1, 4, and 1 from the top. Anna then discards the cards with integers 2, 3, and 5. Bruno discards the top card with integer 3 and the bottom card with integer 1. Thus, their stacks become identical. Each stack has 2 cards, so they will get 2 points.

Anna and Bruno want to maximize their score.

Anna



Bruno



**QUESTION**

Given information about Anna and Bruno's stacks of cards, create a program to find out the maximum score they can get.

**CONSTRAINTS**

1 ≤ A ≤ 5000

1 ≤ B ≤ 5000

The integers written on the cards range from 1 to 1000.

**INPUT**

In the first row, integers A and B are written with a space between them.

In the second row, A integers are written with a space between them. The ith integer, where 1 ≤ i ≤ A, represents the integer written on the ith card from the top of Anna's stack.

In the third row, B integers are written with a space between them. The jth integer, where 1 ≤ j ≤ B, represents the integer written on the jth card from the top of Bruno's stack.

**OUTPUT**

Output the maximum score they can get in one row.

**SCORING CRITERIA**

For 10% of the allocated points, fulfil A ≤ 10 and B ≤ 10.

For 50% of the allocated points, fulfil A ≤ 100 and B ≤ 100.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 5 4  1 2 3 4 5  3 1 4 1 | 2 |

This corresponds to the input in the question.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 6 5  4 1 5 2 3 4  4 5 4 2 3 | 3 |

In this input example, there are two ways they can get 3 points. When Anna discards cards 1, 2, and 3, and Bruno discards cards 2 and 3, their stacks will become identical with a sequence of 4, 5, 4 from the top, so they will get 3 points. Alternatively, Anna discards cards 1, 5, and 4, and Bruno discards cards 4 and 5. Their stacks will become identical with a sequence of 4, 2, 3 from the top, so they will get 3 points.

**3. Night Market**

Taro decided to attend the summer festival held at JOI shrine.

There are N game stalls lined up along the road leading to JOI shrine. Each stall is numbered sequentially from 1 to N, with integers representing how fun the games are and how much time is needed to play them. Stall i has a fun level of Ai, and a play time of Bi.

There is also a fireworks show during the festival, and the largest firework will be released at Time S. Taro wants to see this largest firework.

To enjoy both the games stalls and the fireworks show, Taro plans to arrive at Time 0, and stay until the festival ends at Time T.

Taro chooses k games stalls, where 1 ≤ k ≤ N, and plays the game at each stall. He cannot visit the same stall twice. Starting from the stall with the smallest number, the stalls chosen by Taro are renumbered y1, y2, ..., yk. He will visit Stall i at Time xyi and play games there until Time xyi + Byi.

Taro will visit the stalls from the smallest number to the largest. He cannot play at more than one stall at any time. Time taken to move from stall to stall is negligible.

Once the festival ends at Time T, Taro cannot play at the stalls anymore. Also, he cannot view the fireworks show when he is playing at a stall. However, if he just began or finished playing at a stall at Time S, he can watch the fireworks show from his position.

In other words, the following constraints must be fulfilled.

* y1 < y2 < ... < yk
* xy1, xy2, ..., xyk are integers.
* 0 ≤ xy1 < xy1 + By1 ≤ xy2 < xy2 + By2 ≤ ... ≤ xyk < xyk + Byk ≤ T
* There is no i value where xyi < S < xyi + Byi.

The total fun level Ay1, Ay2, ..., Ayk is M. Taro plans to maximize M as much as he can.

**QUESTION**

Given information about the N games stalls and Times S and T, create a program to find out the maximum value of M.

**CONSTRAINTS**

Number of games stalls: 1 ≤ N ≤ 3000

Time the summer festival ends: 1 ≤ T ≤ 3000

Time the largest firework is released: 0 ≤ S ≤ T

Fun level of Stall i: 0 ≤ Ai ≤ 100000 (= 105)

Play time at Stall i: 1 ≤ Bi ≤ 3000

**INPUT**

In the first row, integers N, T, and S are written with a space between them. This represents that there are N games stalls, the summer festival ends at Time T, and the largest firework is released at Time S.

For the next N rows, in the (i + 1)th row, where 1 ≤ i ≤ N, integers Ai and Bi are written with a space between them. This represents that Stall i has a fun level of Ai and play time of Bi.

Please ensure that Taro visits more than 1 game stall.

**OUTPUT**

Output the maximum value of M in one row.

**SCORING CRITERIA**

For 10% of the allocated points, fulfil N ≤ 20.

For 20% of the allocated points, fulfil S = 0.

For 30% of the allocated points, fulfil at least one of these 2 constraints. No score is allocated for fulfilling both constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 5 20 14  8 9  2 4  7 13  6 3  5 8 | 16 |

In this input example, Taro visits Stall 1 at Time 0, Stall 2 at Time 9, and Stall 4 at Time 14. This will give the maximum value of M.

In this case, M = 8 + 2 + 6 = 16.

**4. Nails**

JOI is hammering some nails into a plank of wood. As shown in the diagram below, JOI lined the nails up into an equilateral triangle, with N nails on each side. The ath line from the top, where 1 ≤ a ≤ N, has a nails. The bth nail from the left, where 1 ≤ b ≤ a, is labeled as Nail (a,b).

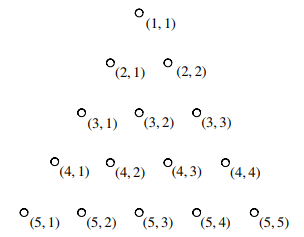


Diagram 1: Arrangement of nails (where N = 5)

Any smaller equilateral triangles formed within the overall triangle that has all sides parallel to the sides of the overall triangle are known as "good equilateral triangles". In other words, a "good equilateral triangle" has its 3 points on the nails (a, b), (a + x, b), and (a + x, b + x), where 1 ≤ a < N, 1 ≤ b ≤ a, and 1 ≤ x ≤ (N − a).

JOI used some rubber bands to mark out "good equilateral triangles".

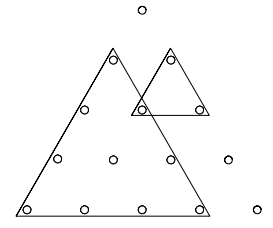


Diagram 2: Example of rubber bands marking out "good equilateral triangles"

**QUESTION**

Given information about the number of nails on one side of the overall triangle N, the number of rubber bands JOI has M, and the "good equilateral triangles" marked out using M rubber bands, create a program to find out the number of nails surrounded by at least 1 rubber band.

**CONSTRAINTS**

Number of nails on one side of the overall triangle: 2 ≤ N ≤ 5000

Number of rubber bands: 1 ≤ M ≤ 500000 (= 5 × 105)

**INPUT**

In the first row, integers N and M are written with a space between them. These represent that there are N nails on one side of the overall triangle, and that JOI has M rubber bands.

For the following M rows, information about the "good equilateral triangles" are written. In the (i + 1)th row, where 1 ≤ i ≤ M, integers Ai, Bi, and Xi, where 1 ≤ Ai < N, 1 ≤ Bi ≤ Ai, and 1 ≤ Xi ≤ N − Ai, are written with a space between them. This represents that the ith rubber band forms a "good equilateral triangle" with its 3 points on the nails (Ai, Bi), (Ai + Xi, Bi), and (Ai + Xi, Bi + Xi).

**OUTPUT**

Output the number of nails surrounded by at least 1 rubber band in one row.

**SCORING CRITERIA**

For 30% of the allocated points, fulfil M ≤ 10000.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 5 2  2 2 1  2 1 3 | 12 |

This corresponds to the "good equilateral triangles" shown in diagram 2. In this example, 12 nails are surrounded by at least 1 rubber band, except for nails (1, 1), (4, 4), and (5, 5).

**5. Festivals in JOI Kingdom**

There are N cities in JOI Kingdom, that are connected to each other via M two-way roads. Citizens move from city to city via these roads.

Many citizens of JOI Kingdom love festivals. Currently, K cities are hosting festivals. However, some citizens dislike the noise of these festivals, and wish to stay away from them if possible.

The king has asked you, a talented programmer, to create a program to find out the best way for citizens who dislike festivals to move away from them, without coming close to a city that is hosting a festival.

**QUESTION**

You are given information about the roads, cities hosting festivals, and Q queries (pairs of current city Si and destination city Ti). For each Query i, create a program to find out, of all the roads connecting Cities Si and Ti, the maximum distance a traveler will be from a city holding a festival during their journey. This is the minimum distance needed to travel between a city along the journey and a city holding the festival.

**CONSTRAINTS**

Number of cities in JOI Kingdom: 2 ≤ N ≤ 100000 (= 105)

Number of roads in JOI Kingdom: 1 ≤ M ≤ 200000 (= 2 × 105)

Number of cities hosting festivals: 1 ≤ K ≤ N

Number of queries: 1 ≤ Q ≤ 100000 (= 105)

The length of Road i: 1 ≤ Li ≤ 1000

**INPUT**

In the first row, integers N, M, K, and Q are written with a space between them. These represent that there are N cities and M roads in JOI Kingdom, K cities are hosting festivals, and there are Q queries. Cities are numbered 1, 2, ..., N.

For the following M rows, information about the roads are written. In the (i + 1)th row, where 1 ≤ i ≤ M, integers Ai, Bi, and Li, where 1 ≤ Ai ≤ N and 1 ≤ Bi ≤ N, are written with a space between them. This represents that Cities Ai and Bi are joined by Road i, which has a length of Li. The two ends of a road must not join to the same city. Also, for any 2 cities p and q, there is only 1 road joining them. Several roads may be taken to travel from one city to another.

For the next K rows, information about the cities hosting festivals are written. In the (i + M + 1)th row, where 1 ≤ i ≤ K, integer Fi, where 1 ≤ Fi ≤ N, is written. This represents that City Fi is hosting a festival. There are no numbers repeated in F1, ..., FK.

For the next Q rows, information about the queries are written. In the (i + M + K + 1)th row, where 1 ≤ i ≤ Q, integers Si and Ti, where 1 ≤ Si ≤ N, 1 ≤ Ti ≤ N, and Si ≠ Ti, are written with a space between them. This represents that Query i has a current city of Si and destination city of Ti.

**OUTPUT**

Output the answer to all queries in Q rows. In other words, in the ith row, of all the roads connecting Cities Si and Ti, output the maximum distance a traveler will be from a city holding a festival during their journey.

**SCORING CRITERIA**

For 10% of the allocated points, fulfil Q = 1.

For 20% of the allocated points, fulfil N ≤ 5000 and Q ≤ 5000.

For 30% of the allocated points, fulfil at least one of these 2 constraints. No score is allocated for fulfilling both constraints.

**INPUT/OUTPUT EXAMPLES**

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

6 cities are connected by 6 roads, and festivals are being held in Cities 1 and 6. There are 3 queries as shown below:

* The first query is to move from City 3 to City 4. The journey going through City 2 will bring the traveler to a distance of 5 from a festival, while the journey going through City 5 will bring the traveler to a distance of 7 from a festival. Therefore, the answer is 7.
* The second query is to move from City 5 to City 2. Going through either City 3 or City 4 will still bring the traveler to a distance of 5 from a festival, which is held in City 2, so the answer is 5.
* The third query is to move from City 1 to City 4. Since a festival is held in City 1, the answer is 0.

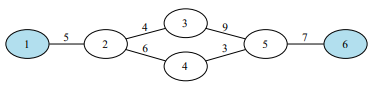


Diagram 1: Input example 1

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 12 17 2 5  1 3 6  1 6 7  2 3 8  2 4 4  2 8 11  2 12 2  3 6 3  3 7 8  3 11 2  4 12 2  5 10 3  6 10 5  8 9 6  8 12 7  9 10 6  11 9 10  12 9 5  8  7  2 6  5 2  1 10  8 9  9 4 | 8  8  11  0  6 |

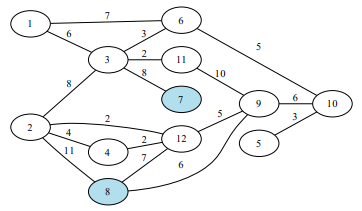


Diagram 2: Input example 2