14th JOI Selection Final Selection Tasks (2015)

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. Railroad Trip**

There are N cities in JOI Kingdom, numbered 1, 2, …, N. There are also N − 1 railroads, numbered 1, 2, …, N − 1. Railroad i, where 1 ≦ i ≦ N − 1, connects City i and City (i + 1).

To ride on JOI Kingdom's railroads, a passenger can buy a paper ticket, or use a train pass.

* Riding on Railroad i by paper ticket costs Ai yen.
* Riding on Railroad i by train pass costs Bi yen. However, a passenger needs to purchase a train pass first. The train pass costs Ci yen. After purchasing a train pass, it can be reused without limit.

Since using the train pass makes things easier, the rail company decides to make traveling by train pass cheaper than traveling by paper ticket. In other words, for i = 1, 2, …, N − 1, Ai > Bi. However, traveling on each railroad requires a different type of train pass. Therefore, a train pass for Railroad i cannot be used to travel on any other railroads.

You plan to travel around JOI Kingdom on vacation. You will start off from City P1, then visit Cities P2, P3, …, PM in sequence. You will travel for M − 1 days. On Day j, where 1 ≦ j ≦ M − 1, you will travel by railroad from City Pj to City (Pj + 1). To do so, you may need to transfer between multiple railroads. You may also visit the same city more than once. All railroads in JOI Kingdom are fast, so it is always possible to travel from one city to any other within 1 day.

Currently, you do not own any train passes. During your journey, you want to spend as little money as possible on travel expenses, which includes buying train passes and riding on all railroads.

**QUESTION**

Given the number of cities in JOI Kingdom, your vacation route, and the costs of riding on each railroad and buying each train pass, create a program to find out the minimum amount of money you can spend on travel expenses.

**INPUT**

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

* N, representing the number of cities in JOI Kingdom.
* M, representing the number of days of travel as M − 1.

In the second row, M integers are written with a space between them.

* P1, P2, . . . , PM, representing the vacation route of traveling from City Pj to City (Pj + 1) on Day j, where 1 ≦ j ≦ M − 1.

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

* Ai, representing the cost to travel on Railroad i by paper ticket.
* Bi, representing the cost to travel on Railroad i by train pass.
* Ci, representing the cost of a train pass for Railroad i.

**OUTPUT**

Output one row with the minimum amount of money you can spend on travel expenses.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 2 ≦ N ≦ 100 000
* 2 ≦ M ≦ 100 000
* 1 ≦ Bi < Ai ≦ 100 000, where 1 ≦ i ≦ N − 1
* 1 ≦ Ci ≦ 100 000, where 1 ≦ i ≦ N − 1
* 1 ≦ Pj ≦ N, where 1 ≦ j ≦ M
* Pj ≠ Pj + 1, where 1 ≦ j ≦ M − 1

**SUBTASKS**

**Subtask 1 [20 points]**

Fulfil the following constraints:

* 2 ≦ N ≦ 1 000
* M = 2
* 1 ≦ Bi < Ai ≦ 1 000, where 1 ≦ i ≦ N − 1
* 1 ≦ Ci ≦ 1 000, where 1 ≦ i ≦ N − 1

**Subtask 2 [30 points]**

Fulfil the following constraints:

* 2 ≦ N ≦ 1 000
* 2 ≦ M ≦ 1 000
* 1 ≦ Bi < Ai ≦ 1 000, where 1 ≦ i ≦ N − 1
* 1 ≦ Ci ≦ 1 000, where 1 ≦ i ≦ N − 1

**Subtask 3 [50 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 4 4  1 3 2 4  120 90 100  110 50 80  250 70 130 | 550 |

In this example, the minimum amount of money you can spend on travel expenses is as follows:

* Buy a train pass to travel on Railroad 2 and Railroad 3. This will cost 80 + 130 = 210 yen.
* On Day 1, travel from City 1 to City 2 using a paper ticket, then travel from City 2 to City 3 using the train pass. This will cost 120 + 50 = 170 yen.
* On Day 2, travel from City 3 to City 2 using the train pass. This will cost 50 yen.
* On Day 3, travel from City 2 to City 3 using the train pass, then travel from City 3 to City 4 using the train pass. This will cost 50 + 70 = 120 yen.

This will cost a total of 210 + 170 + 50 + 120 = 550 for travel expenses.

Since this is the minimum amount, the output is 550.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 8 5  7 5 3 5 4  12 5 8  16 2 1  3 1 5  17 12 17  19 7 5  12 2 19  4 1 3 | 81 |

**2. Cake 2**

JOI and IOI are twins. JOI recently learned how to make pastries and is baking a cake today. IOI is drawn to the smell of the baked cake, so they decided to share it.

The cake is round. They choose a point to make the first cut, then cut the cake into N pieces. Each piece is numbered from 1 to N, in an anti-clockwise direction. In other words, Piece i, where 1 ≦ i ≦ N, is located between Piece (i − 1) and Piece (i + 1). (Piece 0 = Piece N, and Piece 1 = Piece N + 1). Piece i has a size of Ai, but due to a lack of cutting skills, all pieces are of different sizes.

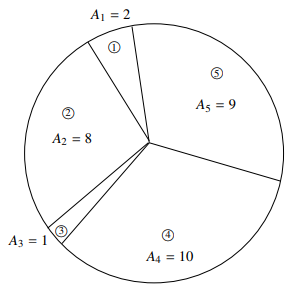


Diagram 1: Example of cake, where N = 5, A1 = 2, A2 = 8, A3 = 1, A4 = 10, A5 = 9.

JOI and IOI divided the cake pieces up according to the following:

1. JOI will choose 1 piece of cake out of N pieces.
2. After that, the twins will take turns choosing 1 piece of cake at a time, starting with IOI. However, they can only choose pieces that are not sandwiched between 2 other pieces. IOI will always pick the largest possible piece, while JOI picks any piece that he likes.

JOI wants to maximize the total size of the cakes he chooses.

**QUESTION**

Given the number of pieces N, and the sizes of each piece of cake, create a program to find out the maximum total size of the cakes JOI chose.

**INPUT**

In the first row, one integer is written.

* N, representing the number of pieces of cake available.

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

* Ai, representing the size of Piece i of cake.

**OUTPUT**

Output one row, with the maximum total size of the cakes JOI chose.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 1 ≦ N ≦ 2 000
* 1 ≦ Ai ≦ 1 000 000 000
* Each Ai is a unique number without repeats.

**SUBTASKS**

**Subtask 1 [15 points]**

Fulfil the following constraint:

* N ≦ 20

**Subtask 2 [45 points]**

Fulfil the following constraint:

* N ≦ 300

**Subtask 3 [40 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 5  2  8  1  10  9 | 18 |

The best way for JOI to choose pieces is as follows:

1. JOI takes Piece 2, which has a size of 8.
2. IOI takes the largest possible piece, Piece 1, which has a size of 2.
3. JOI takes Piece 5, which has a size of 9.
4. IOI takes the largest possible piece, Piece 4, which has a size of 10.
5. JOI takes Piece 3, which has a size of 1.

In the end, the total size of the pieces JOI chose is 8 + 9 + 1 = 18.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 8  1  10  4  5  6  2  9  3 | 26 |

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| 15  182243672  10074562  977552215  122668426  685444213  3784162  463324752  560071245  134465220  21447865  654556327  183481051  20041805  405079805  564327789 | 3600242976 |

**3. JOI Park**

The Olympics will be held in IOI Kingdom in 20XX, and JOI Park is undergoing some upgrades to host the event. There are N plazas in JOI Park, numbered from 1 to N. There are M paths that connect the plazas, numbered from 1 to M. Path i, where 1 ≦ i ≦ M, connects Plaza Ai to Plaza Bi, and has a length of Di. There are multiple ways to travel from one plaza to another.

First, X, an integer greater than 0, is chosen, and all plazas with Di ≤ X measured from Plaza 1 (including Plaza 1 itself) will be connected to each other via new underground tunnels. However, the new underground tunnel connecting Plaza i and Plaza j has a length equal to the minimum length of aboveground paths connecting the two plazas. The cost of constructing an underground tunnel is fixed at C. The total cost needed to build all tunnels is C × X.

Next, all aboveground paths that used to connect the plazas now joined by underground tunnels are demolished. Demolition of these paths does not cost any money.

Finally, all remaining paths are renovated. A path of length d has a renovation cost of d.

There are no underground tunnels before the construction plan begins. The committee in-charge wants to minimize the cost of these upgrades.

**QUESTION**

Given information about the plazas in JOI Park and the cost of constructing underground tunnels, create a program to find out the minimum cost needed to upgrade JOI Park.

**INPUT**

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

* N, representing the number of plazas.
* M, representing the number of paths.
* C, representing the cost of constructing an underground tunnel.

For the next M rows, in the ith row, where 1 ≦ i ≦ M, three integers are written with a space between them.

* Ai and Bi, representing two plazas that are connected by Path i.
* Di, representing the length of Path i.

**OUTPUT**

Output one row, with the minimum cost needed to upgrade JOI Park.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 2 ≦ N ≦ 100 000
* 1 ≦ M ≦ 200 000
* 1 ≦ C ≦ 100 000
* 1 ≦ Ai ≦ N, where 1 ≦ i ≦ M
* 1 ≦ Bi ≦ N, where 1 ≦ i ≦ M
* Ai ≠ Bi, where 1 ≦ i ≦ M
* (Ai,Bi) ≠ (Aj,Bj) and (Ai,Bi) ≠ (Bj,Aj), where 1 ≦ i < j ≦ M
* 1 ≦ Di ≦ 100 000, where 1 ≦ i ≦ M
* Ensure that there are multiple ways to travel from one plaza to another, given the input data.

**SUBTASKS**

**Subtask 1 [15 points]**

Fulfil the following constraints:

* N ≦ 100
* M ≦ 200
* C ≦ 100
* Di ≦ 10, where 1 ≦ i ≦ M

**Subtask 2 [45 points]**

Fulfil the following constraints:

* N ≦ 100
* M ≦ 4 000

**Subtask 3 [40 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

|  |  |
| --- | --- |
| INPUT 1 | OUTPUT 1 |
| 5 5 2  2 3 1  3 1 2  2 4 3  1 2 4  2 5 5 | 14 |

In this input example, if X = 3, all plazas with Di ≤ 3 measured from Plaza 1 (Plaza 1, Plaza 2, Plaza 3) will be connected to each other via new underground tunnels. The total cost of upgrades is 2 × 3 + 3 + 5 = 14. This is the minimum output.

|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 5 4 10  1 2 3  2 3 4  3 4 3  4 5 5 | 15 |

In this input example, if X = 0, the minimum output is 15.

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| 6 5 2  1 2 2  1 3 4  1 4 3  1 5 1  1 6 5 | 10 |

In this input example, if X = 5, all plazas with Di ≤ 5 measured from Plaza 1 (all plazas) will be connected to each other via new underground tunnels. The minimum output is 10.

**4. Ball**

IOI Kingdom is holding a ball to celebrate the birthday of Princess JOI.

N royal people are invited to the ball. N is an odd number. The guests are numbered from 1 to N. Each guest has an integer representing their dance skill, so Guest i, where 1 ≦ i ≦ N, has a dance skill of Di.

During the ball, the (N + 1) people, which includes Princess JOI, pair off to dance. The king of IOI Kingdom wants good dancers to help beginners, so the dance pairs are decided as follows:

* First, all N guests line up in one row.
* The following process is repeated until there is only 1 guest left in the row:
  + The dance skills of the first 3 guests in the row are compared.
  + The guest with the highest dance skill of the 3 is labeled A. In the case where more than 1 guest has the highest dance skill, the guest with the smaller number is labeled A.
  + The guest with the lowest dance skill of the 3 is labeled B. In the case where more than 1 guest has the lowest dance skill, the guest with the larger number is labeled B.
  + A and B will leave the row to form a dance pair.
  + The remaining guest will go to the back of the row.
* The final guest left in the row pairs up with Princess JOI.

Guest 1 to Guest M, where 1 ≦ M ≦ N − 2, have already determined where to stand in the row. The remaining (N − M) guests are inserted into the row at the king's orders.

Princess JOI is a beginner dancer, so the king wishes to pair her to a guest with the highest possible dance skill.

**QUESTION**

Given the dance skill of each guest and the position of M guests in the row, create a program to find out the maximum dance skill of Princess JOI's dance partner.

**INPUT**

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

* N, representing the number of guests at the ball.
* M, representing the number of guests who have determined their position in the row.

For the next M rows, in the ith row, where 1 ≦ i ≦ M, two integers are written with a space between them.

* Di, representing the dance skill of Guest i.
* Pi, representing the position of Guest i in the row.

For the next (N − M) rows, in the ith row, where 1 ≦ i ≦ N − M, one integer is written.

* Di+M, representing the dance skill of Guest (i + M).

**OUTPUT**

Output the maximum dance skill of Princess JOI's dance partner in one row.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 3 ≦ N ≦ 99 999
* N is an odd number.
* 1 ≦ M ≦ N − 2
* 1 ≦ Di ≦ 1 000 000 000, where 1 ≦ i ≦ N
* 1 ≦ Pi ≦ N, where 1 ≦ i ≦ M
* Pi ≠ Pj, where 1 ≦ i < j ≦ M

**SUBTASKS**

**Subtask 1 [8 points]**

Fulfil the following constraint:

* N ≦ 9

**Subtask 2 [16 points]**

Fulfil the following constraint:

* N ≦ 19

**Subtask 3 [44 points]**

Fulfil the following constraint:

* N ≦ 1 999

**Subtask 4 [32 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

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

The positions of 3 guests have been determined.



The number in the brackets represents the dance skill of the guest.

The head of the line is on the left side.

The guests are lined up according to the sequence: 5-1-4-6-2-3-7.



Diagram showing all guests in a row.

In this situation, the following will take place:

* Considering the first 3 guests in the row (Guests 5, 1, 4), Guest 4 who has the highest dance skill will pair with Guest 5 who has the lowest dance skill. Guest 1 will go to the back of the row.
* Next, considering the first 3 guests in the row (Guests 6, 2, 3), there are two guests, Guests 3 and 6, with the highest dance skill. Guest 3 has the smaller number, so they will pair with Guest 2 who has the lowest dance skill. Guest 6 will go to the back of the row.
* Next, considering the first 3 guests in the row (Guests 7, 1, 6), Guest 7 who has the highest dance skill will pair with Guest 1 who has the lowest dance skill. Guest 6 will go to the back of the row.
* Guest 6 is the final guest left in the row and pairs up with Princess JOI.

Guest 6 has a dance skill of 8. This is the maximum possible dance skill of Princess JOI's dance partner.

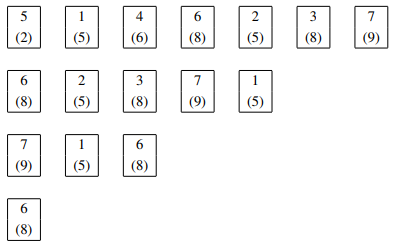


Diagram showing changes in the row

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

The sequence of guests does not matter, Princess JOI will be paired with Guest 2.

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| 7 2  32 4  27 6  37  41  41  30  27 | 37 |

**5. Ramparts**

History professor JOI is doing research on an ancient civilization, IOI Kingdom.

According to past studies, IOI Kingdom is divided into a grid with H columns and W rows of squares. The capital city of IOI Kingdom is surrounded by ramparts for defense.

The ramparts are given a value representative of its size. Ramparts with a size of s, where s ≧ 3, take up a square area of s × s, closing off a square with area [(s − 2) × (s − 2)] from the surroundings.

Professor JOI's research tells him that the ramparts have a minimum size of L. He also knows that there are some squares on the grid which do not contain ramparts.

To further his research, professor JOI wants to find out how many ways the ramparts of IOI Kingdom could have been arranged.

**QUESTION**

Given the size of IOI Kingdom, the minimum size of ramparts, and information about the squares on the grid which do not contain ramparts, create a program to find out how many ways the ramparts of IOI Kingdom could have been arranged.

**INPUT**

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

* H, representing the number of columns in IOI Kingdom's grid.
* W, representing the number of rows in IOI Kingdom's grid.
* L, representing the minimum size of ramparts.
* P, representing the number of squares on the grid which do not contain ramparts.

For the next P rows, in the ith row, where 1 ≦ i ≦ P, two integers are written with a space between them.

* Ai, representing the position of Square i, which does not contain ramparts, from the top.
* Bi, representing the position of Square i, which does not contain ramparts, from the left.

**OUTPUT**

Output one row, with the number of ways the ramparts of IOI Kingdom could have been arranged.

**CONSTRAINTS**

All input data must fulfil the following constraints:

* 1 ≦ H ≦ 4 000
* 1 ≦ W ≦ 4 000
* 3 ≦ L ≦ H and 3 ≦ L ≦ W
* 0 ≦ P ≦ 100 000
* 1 ≦ Ai ≦ H, where 1 ≦ i ≦ P
* 1 ≦ Bi ≦ W, where 1 ≦ i ≦ P
* (Ai,Bi) ≠ (Aj,Bj), where 1 ≦ i < j ≦ P

**SUBTASKS**

**Subtask 1 [4 points]**

Fulfil the following constraints:

* H ≦ 500
* W ≦ 500

**Subtask 2 [16 points]**

Fulfil the following constraint:

* 0 ≦ P ≦ 10

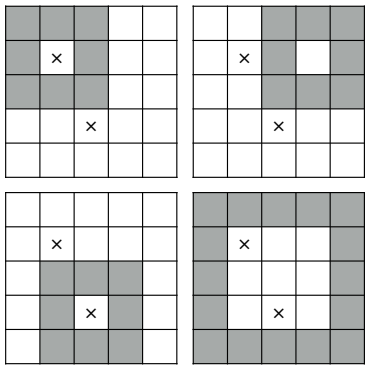
**Subtask 3 [80 points]**

No additional constraints.

**INPUT/OUTPUT EXAMPLES**

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

In this input example, there are 4 possible ways the ramparts of IOI Kingdom could have been arranged. Take note that the squares marked with × are known to not contain ramparts.



|  |  |
| --- | --- |
| INPUT 2 | OUTPUT 2 |
| 7 8 4 3  2 2  3 7  6 5 | 13 |

|  |  |
| --- | --- |
| INPUT 3 | OUTPUT 3 |
| 4000 4000 1234 4  1161 3028  596 1892  3731 2606  702 1530 | 7050792912 |