Advanced Algorithms and Datastructures

Challenges and Solutions

Last modified:
28th March 2018

Contents

1 | Introduction & Rules

Over the duration of this course challenges will continually become available. Solving these challenges will allow you to accumulate points. These points will, in the end, translate directly into your end grade. There are however some rules, which will be explained below.

Submitting: For submitting your solutions you will have to upload your programs to Themis. Each submission should be made individually and before the deadline, as the submission tool will close afterwards.

Grading: You will obtain points for the amount of test cases you manage to pass. The points will be distributed equally over all available test cases for a challenge. I.e. if a challenge has 10 test cases and a maximum reward of 40 points, then each passed test case rewards you 4 points.

Each point directly translates into .01 grade point. Meaning that a challenge of 40 points may increase your grade by .4, potentially. Although, you will be able to accumulate more than 600 points through all the challenges, only the first 600 points will influence your end grade.

After the deadline closes, the number of points earned for each challenge is given by the percentage of tests passed by the last submission. The TA's will not open each submission individually so the quality of code or the implementation tricks do not matter. The verdict given by Themis will be considered final.

Libraries & Plagiarism: We will never ask you to reinvent the wheel or ignite a burning candle. You are advised to make use of existing libraries where you see fit. This only applies to the standard libraries, though. Third-party dependencies are prohibited. Additionally, your submissions will be checked for plagiarism, so make sure your submission contains your own work only. The cases of plagiarism will be brought further than a simple dissmisal from the course.

Bonuses: In case you manage to solve a challenge in two or more different languages the maximum amount of points you may obtain from that challenge will increase with 10 %. This is under the condition that in all languages all test cases are passed. Besides the language bonus, you can receive two extra bonuses:

1. **First Blood**: The first student to submit a passing solution (irrelevant of the language) receives a bonus of 25% per challenge.
2. **Fastest Lap**: The submission (per each individual language) that passes with the least amount of time earns its student a bonus of $v * 0.25 * min((n/10), 1)$ where v is the value in points of the challenge and n is the number of students that have passed solutions for the challenge.

Furthermore, points are awarded for completing HackerRank challenges. The amount of points awarded is one-tenth the amount the hacko's the challenge is worth. Through this bonus you may obtain a maximum of 25 points.

I | Challenges

2 | Aku (AKU) - 40 points

Your friend Samurai Jack, a young prince, has received a magic katana in order to defeat and imprison the supernatural shape-shifting demon Aku. The demon Aku is so supernatural that now he has transformed itself into a natural number! Jack has to very quickly perform a series of complicated moves in order to defeat Aku.

Initially, Jack sees on the field (which is a sequence of integers) only the number n — the number in which Aku hides. Then he has to battle in the following way: each move, Jack uses his katana to split any number $x > 1$ on the field, into the numbers $\left\lfloor \frac{x}{2} \right\rfloor, x \pmod 2, \left\lfloor \frac{x}{2} \right\rfloor$ which appear sequentially in the same position. In this way, Jack has to dissect each part of Aku until all the numbers on the field are either 0 or 1. Notice that the final state of the field is invariant under the order of the operations performed.

Now, in order to make the final move, he needs to count the total number of 1s in the range of l to r (1-indexed). Unfortunately Jack cannot do it sufficiently fast and asks you for help.

2.1 | Input

The first line contains three integers n, l, r ($0 \leq n < 2^{50}$, $0 \leq r - l \leq 10^5$, $r \geq 1$, $l \geq 1$) — the initial number of Aku and the range from l to r. It is guaranteed that r is not greater than the length of the final list.

2.2 | Output

Print the number of 1s in the given range on final state of the field.

2.3 | Example

Input	Output
`7 2 5`	`4`
`10 3 10`	`5`

2.4 | Note

Consider the first example:

$[7] \to [3,1,3] \to [1,1,1,1,3] \to [1,1,1,1,1,1,1]$ The elements on the positions from 2 to 5 are $[1,1,1,1]$ so the answer is 4.

$[10] \to \ldots \to [1,0,1,1,1,0,1,0,1,0,1,1,1,0,1]$ The elements on the positions from 3 to 10 are $[1,1,1,0,1,0,1,0]$ so the answer is 5.

2.5 | Restrictions

Remember that the `BufferedReader` in Java is faster than the `Scanner`, this may mean the difference between reaching the time limit and passing the tests.

3 | Multiplication Tables (TABLE) - 40 points

While you were learning the multiplication table in the primary school, Jaap was having some fun on his own.

Jaap draw a $n \times m$ multiplication table, where the element on the intersection of the i-th row and j-th column equals $i \cdot j$, given that the rows and the columns are 1-indexed. Jaap's PhD supervisor then asked him: what number in the table is the k-th largest number? Jaap answered correctly within two seconds. Can you do it faster?

Consider the given multiplication table. If you write out all the $n \cdot m$ numbers from the table in non-decreasing order, then the k-th number you write out is called the k-th largest number on the table.

3.1 | Input

The single line in the input contains three integers: n, m and k ($1 \leq n, m \leq 10^5$; $1 \leq k \leq n \cdot m$).

3.2 | Output

Print the k-th largest number in a $n \times m$ multiplication table.

3.3 | Example

Input	Output
`2 2 2`	`2`
`2 3 4`	`3`
`1 10 5`	`5`

3.4 | Note

The 2×3 multiplication table looks like this:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$$

3.5 Restrictions

Remember that the `BufferedReader` in Java is faster than the `Scanner`, this may mean the difference between reaching the time limit and passing the tests.

4 | Game (GAME) - 40 points

Anna doesn't like boredom. Whenever she gets bored, she invents a new game to play. On this hot day of summer she came up with a game and decided to play it.

You are given a sequence s of integers of length n. The game is played in several steps. In one step you can choose one element s_k of the sequence and delete it (a single copy); together with it, *all* the elements equal to $s_k - 1$ and $s_k + 1$ also must be deleted. This whole step brings the player s_k points.

Anna is an overachiever, so she wants to end the game with as many points as possible. Help her.

4.1 | Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the length of the initial sequence. The second line contains n integers $s_1, s_2, \ldots, s_n$ — the given sequence ($1 \leq s_i \leq 10^5$).

4.2 | Output

Print the maximum number of points Anna can earn.

4.3 | Example

Input	Output
`3` `1 2 3`	`4`
`9` `1 2 1 3 2 2 2 2 3`	`10`

4.4 | Note

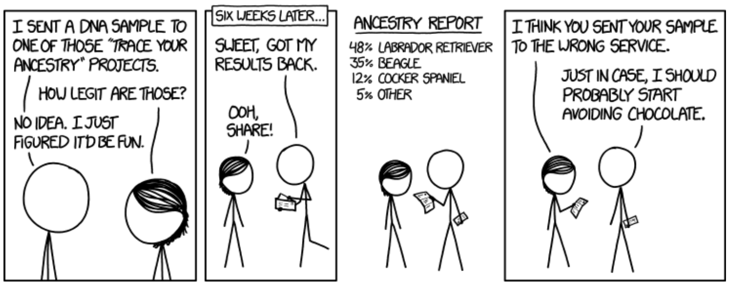
Consider the third test example. At first step we need to choose any element equal to 2. After that step our sequence looks like this $[2, 2, 2, 2]$. Then we do 4 steps, on each step we choose any element equals to 2. In total we earn 10 points.

4.5 | Restrictions

Remember that the `BufferedReader` in Java is faster than the `Scanner`, this may mean the difference between reaching the time limit and passing the tests.

5 | Ancestors (ANCESTORS) - 40 points

The family of Jan de Vries is very numerous, having exactly N members. Each member knows who is his direct ancestor, but a few who don't have ancestors in the de Vries family. Jan, being very curious, has created a portfolio of his family, that is, he numbered each member with a distinct number from 1 to N and he also added who is the direct ancestor of each member (if it exists). He tries to mollify his obstinate curiosity by asking M questions of the type: "Who is the Pth ancestor of the member Q?".



Can you assist him in answering them quickly?

5.1 | Input

The first line of the input contains the numbers N and M ($1 \leq N \leq 250000$, $1 \leq M \leq 300000$). The second line contains N integers A_i, representing the ancestors of the members i. If a member doesn't have an ancestor, then $A_i = 0$. The following M lines represent the queries (Q_i, P_i) as escribed above.

5.2 | Output

The output should on each line the M answers to the queries. The answer constists of the number of the ancestor (or 0 if the identity of the ancestor is unknown).

5.3 | Example

Input	Output

`13 7`
`0 1 2 2 4 1 6 0 8 8 10 10 12`
`5 2`
`3 2`
`7 1`
`1 3`
`13 3`
`9 2`
`11 1`
	`2`
	`1`
	`6`
	`0`
	`8`
	`0`
	`10`

5.4 | Restrictions

Remember that the `BufferedReader` in Java is faster than the `Scanner`, this may mean the difference between reaching the time limit and passing the tests.

6 Very Nice String (STRING) - 20 points

You got a very very nice string s of lowercase letters as a present. The string of characters is 1-indexed, with $\|s\|$ being the length of the given string.

You didn't like the present very much so you decided to change it. You plan to spend m days performing the following transformations on the string — each day you choose an integer a_i and reverse a piece of string (a segment) from position a_i to position $\|s\| - a_i + 1$ (see restrictions for a_i).

You face the following task: determine what your string will look like after m days.

6.1 Input

The first line of the input contains your string s of length from 2 to $2 \cdot 10^5$ characters, consisting of lowercase Latin letters.

The second line contains a single integer m $(1 \leq m \leq 10^5)$ the number of days when you change the string.

The third line contains m queries a_i $(1 \leq a_i; 2 \cdot a_i \leq \|s\|)$ the position from which you start transforming the string on the i-th day.

6.2 Output

In the first line of the output print what your string s will look like after m days.

6.3 Example

Input	Output

`abcdef`
`1`
`2`
	`aedcbf`

`vwxyz`
`2`
`2 2`
	`vwxyz`

7 | Library (LIBRARY) - 20 points

Libraries have in their collections many books, journals and other things. In order to keep track them, every item has its unique number.

After an mischievous hack attack on our library system, you were surprised to find that the items are not numbered increasingly anymore. Moreover, more items may share the same identifier. You have decided to renumber the items with integers from 1 to n. Changing the identifier of a book or journal takes a lot of manual power, so you want to change the least amount of identifiers.

You have been given the system after the attack, consisting of the listing of n identifiers. Change the identifiers such that every item has a unique unique one, using only integers from 1 to n. If more permutations can be created by changing the same minimal amount of numbers, generate the **smallest lexicographical** one.

7.1 | Input

The first line contains n — the number of items ($1 \leq n \leq 10^5$). The second line contains n numbers a_i, ($1 \leq a_i \leq 10^5$) — the identifiers of the items after the attack.

7.2 | Output

Output the list of n identifiers — the corrected list of item identifiers in the same order as it was given after the attack. If there are multiple possible answers, print the **smallest lexicographical** one.

7.3 | Example

Input	Output
`3` `3 1 2`	`3 1 2`
`5` `3 3 3 2 2`	`1 3 4 2 5`
`2` `2 100`	`2` `2 1`

8 | Ferry (FERRY) - 60 points

Daniel is planning a biking tour across Europe. He will have to take n ferries during the trip and he already knows that he will have to spend c_i cents (value) for the i-th ferry. *Note that Daniel will take the ferries in the order it is given the input.*

He knows that the ferrymen only accept and return coins of 1 cent or 1 euro. At the beginning of the trip, Daniel has m 1-cent coins and a sufficiently large amount of 1-euro coins (consider them to be of infinite amount).

Since Daniel cares about the total happiness in the world, he will try to minimize the *discontent* of the ferrymen by trying to pay change-free as much as possible. More precisely, if the i-th ferryman has to give Daniel x coins back, his discontent equals $x \cdot d_i$. The ferrymen always give change with the least amount of coins possible assuming that they have an infinite amount of each of the two types.

Daniel wants to pay to the ferrymen in a way such that the total discontent caused to the world is as small as possible. Note that Daniel can always pay his tickets because he has an infinite amount of 1-euro coins. Moreover, Daniel can use coins of both type that he received as change from the previous ferrymen.

8.1 | Input

The first line contains two integers n and m, ($1 \leq n \leq 3 \cdot 10^5$, $0 \leq m \leq 10^9$) representing the number of ferries on Daniel's trip and the amount of 1-cent coins at it's beginning.

The next two lines contain the prices for each of the n ferry tickets and the discontent factors d_i of the ferrymen ($1 \leq c_i, d_i \leq 10^5$).

8.2 | Output

Print the minimal amount of total discontent that Daniel can cause to the ferrymen.

8.3 | Example

Input	Output

`5 10`
`121 31 183 493 100`
`1 1 1 1 1`
	`103`

`4 10`
`100 50 50 50`
`1 3 2 4`
	`250`

`5 69`
`107 84 161 234 300`
`5 4 3 2 1`
	`181`

9 | Walk (WALK) - 40 points

For his last trick, the magician pulled out of his hat a tree rooted at vertex 1, of size N, with digits drawn on every vertex. As part of the performance, he asks you the following interesting questions:

The magician gives you a prime number P, co-prime to 10, and a list of Q questions. In each question, you are given 2 vertices a and b of the tree. The vertex a is always be an ancestor of b, unless $a = b$. Now, the magician asks you if the number constructed by writing out and appending the digits on the path from a to b, is divisible by P.

You are bored by his dull tricks and want to end his show as fast as possible by answering the questions.

Hint: The challenges for this week are focused on the concepts of Hashing and Balanced Search Trees.

9.1 | Input

The first line contains the size of the tree ($1 \leq N \leq 2 \cdot 10^5$), the prime number ($2 \leq P \leq 2 \cdot 10^7$, $\gcd(P, 10) = 1$) and the number of questions $1 \leq Q \leq 2 \cdot 10^5$.
The second line contains N digits, where the i-th one is the digit drawn on the i-th vertex.
The third line contains $N - 1$ integers where the i-th one denotes the parent of the vertex $i + 1$.
The next Q lines describe the questions in the format (a_i, b_i).

9.2 | Output

Answer the queries with `"YES"` or `"NO"` on separate lines (without quotes).

9.3 | Example

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>7 3 3
6 1 2 5 2 1 2
1 2 2 1 5 5
1 3
1 4
5 7</pre></td><td><pre>YES
YES
NO</pre></td></tr>
</table>

9.4 | Explanation

The numbers constructed for the queries are 612, 615 and 22, where only the first 2 are divisible by 3.

10 | Quiz (QUIZ) - 60 points

During the history class "Galaxy Wars", you study N wars that developed during the years $[L_i, R_i]$. Now, the meticulous professor has given out a quiz with Q questions to test your memorisation of all the wars in the history.

Each question contains two integers K and X. You need to find the length of the K-th shortest war that occured during the year X. If less than K wars occured during the year X, write down -1.

We say that the war $[L_i, R_i]$ occured during the year X if $L_i \leq X \leq R_i$. We define the duration of a war as the number of years during which it occured: $R_i - L_i + 1$.

Hint: The challenges for this week are focused on the concepts of Hashing and Balanced Search Trees.

10.1 | Input

The first line contains the number of wars ($1 \leq N \leq 2 \cdot 10^5$). Each of the next N lines contains the starting and the ending year of a war ($1 \leq L_i, R_i \leq 10^9$). The next line contains the number of questions ($1 \leq Q \leq 2 \cdot 10^5$) on the quiz and the following Q lines the questions in the format (K, X), ($1 \leq K \leq N$, $1 \leq X \leq 10^9$).

Quick, the time is ticking!

10.2 | Output

Write down the answer to every question on a new line.

10.3 | Example

Input	Output
5 2 4 2 6 4 7 8 16 2 19 3 1 3 2 8 4 5	3 18 -1

11 | Color (COLOR) - 40 points

An undirected graph $G = (V, E)$ with N vertices and M edges is given to you. A *cool coloring* of the graph is an assignment of either the red or the blue color to each vertex such that the end points of any edge are not of the same color.

Since you like the color red, you want to find a *cool coloring* such that the maximum number of red vertices is attained.

11.1 | Input

The first line contains T $(1 \leq T \leq 3)$, the number of tests in one file. The next lines describe each graph. A graph's description starts with a line containing N and M $(1 \leq N \leq 100000, 1 \leq M \leq 200000)$. Each of the next M lines contains two integers a_i and b_i denoting an undirected edge between the node a_i and b_i $(1 \leq a_i, b_i \leq N)$.

11.2 | Output

For every graph, output the maximum number of red vertices possible in a *cool coloring*. If there is no *cool coloring* possible, output -1. Give the answer for each graph on a new line.

11.3 | Example

Input	Output

`1`
`3 2`
`1 2`
`2 3`
	`2`

12 | Bank Robbery (BANK) - 40 points

Bonny and Clyde are on their next adventure. This time they are robbing a bank in a graph $G = (V, E)$ with N vertices and M edges. The graph has O banks in some subset of distinct vertices. Bonny and Clyde are aware of the Graph Police that will catch them no matter what in P minutes after a robbery, unless they can make it to their hiding vertex B in less than P minutes. Assuming that the revenue of robbing any bank is the same, find the minimum amount of time that it will take Bonny and Clyde to rob any bank and get to their hiding point before the Graph Police catches them. Initially they are located in the vertex A.

Consider that it takes one minute to traverse one unit of distance.

12.1 | Input

The first line of the input contains the integers N, M, O, P ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^6$, $1 \leq O \leq N-1$, $1 \leq P \leq 10^9$). The second line contains O integers giving the indices of vertices that have a bank. The next M lines describe the graph edges with three integers a_i, b_i, c_i meaning that there is an undirected edge between a_i, b_i of length c_i, ($1 \leq c_i \leq 500$). The last line contains the indices A, B of the initial and the desired last vertex of the adventure.

12.2 | Output

Output the least amount of time it would require to rob a bank and successfully hide. If it is not possible, output -1 instead.

12.3 | Example

Input	Output
7 3 1 6 1 4 7 1 3 5 7 6 1 3 6 2	-1
6 6 2 3 4 3 1 2 1 2 3 2 2 4 4 3 5 2 4 5 1 5 6 1 1 6	7

13 | Spaceship (SPACESHIP) - 40 points

Houston, we have a problem!

The first spaceship to Mars consists of N blocks and gates that connect them to form a **tree** structure of N vertices. There is an unavoidable collision of the block 1 with a small interplanetary meteorite that will lead to a pressure drop on all the blocks of the spaceship to which the path from block 1 consists of open gates.

For each gate you know whether you are able to close it or not. Your goal is to find the minimum number of gates that you have to close such that you would lose pressure on a maximum of K blocks of the spaceship.

13.1 | Input

The first line contains N and K, $(1 \leq K \leq N \leq 10^5)$. The next $N-1$ lines contain three integers a, b and c which represent a gate between the block a and b with c being 1 if you are able to close the gate and 0 otherwise.

13.2 | Output

If you cannot accomplish the task and the spaceship will explode, output -1. Otherwise, print the minimum number of gates that you have to close.

13.3 | Example

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>6 4
3 5 1
1 2 0
2 3 1
3 4 0
5 6 1</pre></td><td><pre>1</pre></td></tr>
<tr><td><pre>4 2
1 4 0
1 2 0
3 4 1</pre></td><td><pre>-1</pre></td></tr>
<tr><td><pre>4 4
1 4 0
4 3 0
3 2 0</pre></td><td><pre>0</pre></td></tr>
</table>

14 | A sock problem (SOCKS) - 40 points

Rick has a sock problem, after the laundry day he is not able to match up his socks into pairs. The type of each sock is identified as a string of lowercase letters. To solve his problem, Rick has build a machine that allows him to pick a sock and transform it to another type by choosing an integer $k \geq 0$ and shifting each letter of the string k times on the alphabet cycle ($a \to b$, $b \to c$, $\ldots$, $z \to a$).

You are given the types of N socks. For each sock determine whether it can be transformed to match up with another sock from the pile.

14.1 | Input

The first line consists of the number N of socks. The next N lines consist of strings s_i denoting the type of each sock $\sum_{i=1}^{N} length(s_i) \leq 10^5$.

14.2 | Output

Output N lines answering Rick's questions. Print 1 if the i-th sock can be paired after a transformation and 0 otherwise.

14.3 | Example

Input	Output
3 a ab cd	0 1 1
4 bbc xyz abc bbc	1 1 1 1
5 n m nm mn on	1 1 1 0 1

15 | Lan Party (LAN) - 60 points

The company you are working at organizes a big LAN party. For the party you will use the company network which consists of N peer-to-peer connected computers that form an undirected **tree** with N vertices.

Everybody knows that a good LAN party has a good and stable network. In order to construct a subnetwork of maximal stability, you are allowed to perform the following splitting operation a maximum of $N-1$ times:

- Delete a connection between two computers and continue with only one of the two formed connected components, disregarding the second one.

The stability of the final network will be computed as $\sum_{v \in C} s[N(v)]$ where C is the final component $N(v)$ is the number of neighbours of v and the array s is given in the input.

Contribute to the great party and find what is the most stable subnetwork that you can build. Notice that it is possible that the most stable subnetwork could consist of just a single computer!

15.1 | Input

The first line of the input consists of the number N of computers in the network ($1 \leq N \leq 10^5$). The second line consists of the 0-indexed array s of integers ($-10^9 \leq s_i \leq 10^9$). The last $N-1$ lines consist of two integers a and b denoting a connection between the computer a and b in the original network.

15.2 | Output

Output a single integer — the maximum stability of a network after applying the splitting operation at most $N-1$ times.

15.3 Example

Input	Output

`4`
`10 1 2 3`
`1 2`
`1 3`
`2 4`
	`10`

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

`4`
`-10 1 -2 -3`
`1 2`
`2 3`
`3 4`
	`2`

16 | Boxes (BOXES) - 40 points

Freshly moved to Manhattan, Rachel unpacked all of her N cubic boxes of various sizes ($a \times a \times a$). She wants to pack away the empty moving boxes and will proceed to do so by following these simple steps:

- Rachel finds a box of size a.
- She chooses to either put this box in another box found so far (of size $b \leq a$) *or* put the box near the boxes processed so far.

Note that Rachel would not put two boxes side by side in a another box even if it was possible (ex. boxes of size 4 and 5 in a box of size 9), but will always stack them one into another (ex. box of size 5 in box of size 9 and then the box of size 4 in the box of size 5).

Find out what is the minimal amount of boxes that she will have to carry downstairs considering that at once she can carry a box with an arbitrary amount of boxes stacked inside it.

16.1 | Input

The first line of the input contains the number of boxes N that Rachel has in her appartment ($1 \leq N \leq 10^5$). The next line contains N integers a_i — the sizes of the boxes given in the order that she finds them ($1 \leq a_i < 10^7$).

16.2 | Output

Output one integer — the minimal number of boxes that she will have to carry downstairs.

16.3 | Example

Input	Output
`9` `8 11 9 10 7 2 4 2 4`	`3`

17 | Manhattan (MANHATTAN) - 40 points

There are a lot of buildings in the city of New York. The city map is represented by a matrix of size $N \times M$ with `'0'` and `'1'` elements where every cell represents a 1×1 region of the city. The matrix cell contains a '1' if that region is occupied by a building and a '0' otherwise. A building is determined by a submatrix consisting of only `'1'`-cells. It is guaranteed that every 1-cell is part of a valid building and any two buildings are separated (don't have touching edges or corners).

Chandler's boss gave him the task to find the best location in town to park an ice-cream van, in a region of a city which is not occupied by any building. The location is optimal if the sum of the distances between the ice-cream van and all the buildings is minimized. The distance between a building and a region C is defined as the minimum of the Manhattan distances between each region occupied by the building and the region C.

Chandler loves ice cream, but hates his job, help him with this task!

17.1 | Input

The first line contains N and M, the sizes of the given map ($1 \leq N, M \leq 1000$). The next N lines describe the binary valued matrix representing the map.

17.2 | Output

The output should consist of two integers representing the row and the column of the optimally chosen location of the ice-cream van. If multiple locations are optimal, choose the one with smallest row index. If there are still multiple solutions, the one with smallest column index will be chosen.

17.3 | Example

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>7 7
0 1 0 1 0 1 1
0 1 0 1 0 1 1
0 0 0 1 0 0 0
0 0 0 1 0 0 0
0 0 0 1 0 0 0
1 1 0 1 0 1 1
1 1 0 1 0 1 1</pre></td><td><pre>2 3</pre></td></tr>
<tr><td><pre>4 4
0 0 1 1
0 0 1 1
0 0 1 1
0 0 0 0</pre></td><td><pre>1 2</pre></td></tr>
</table>

18 | Friends (FRIENDS) - 40 points

Ai and her friend AI are planning to move to the city of Groningen. The two have already picked K rooms in the city from which they can choose from. Since Ai and AI are very close friends, they want to find two houses in the city which are closest to one another.

Consider the city of Groningen being represented by an undirected graph of N vertices and M edges of length 1. The list of the houses they chose represents the distinct nodes in the graphs where the houses are located.

18.1 | Input

The first line of the input contains N, M and K — the number of vertices and edges in the city of Groningen ($1 \leq N, M \leq 10^5$, $1 \leq K \leq N$). The second line contains K integers representing the list of the nodes where the houses are located. The next M lines contain pairs of nodes (a, b) denoting an edge in the graph.

18.2 | Output

Print a single integer — the shortest path between Ai and AI after they move in Groningen given that they choose their accomodation optimally.

18.3 | Example

Input	Output

`5 6 2`
`1 2`
`2 3`
`2 4`
`3 4`
`1 4`
`3 5`
`1 5`
	`3`

	`1 2`

II | Reference Solutions