HACK WITH INFY
SAMPLE QUESTIONS

Infosys®
Navigate your next

1. EASY

You're given an array A of n integers and q queries. Each query can be one of the following two types:

- **Type 1 Query:** (1, l, r) - Replace A[i] with (i-l+1)*A[l] for each index i, where l <= i <= r.
- **Type 2 Query:** (2, l, r) - Calculate the sum of the elements in A from index l to index r.

Find the sum of answers to all type 2 queries. Since answer can be large, return it modulo 109+7.

Sample Input 1
7
1
4
5
1
6
7
8
5
1 1 6
1 1 5
2 5 5
2 3 4
2 3 3

Sample Output 1
60

Sample Output Description 1
Here, n = 7
A = [1, 4, 5, 1, 6, 7, 8]
q = 5
queries = [[1, 1, 6], [1, 1, 5], [2, 5, 5], [2, 3, 4], [2, 3, 3]]

for query 1 -> (1, 1, 6)
Applying the operation on subarray from index 1 to 6, A becomes, A = [1, 4, 8, 12, 16, 20, 24]

for query 2 -> (1, 1, 5)
Applying the operation on subarray from index 1 to 5, A becomes, A = [1, 4, 8, 12, 16, 20, 24]

for query 3 -> (2, 5, 5)
calculate sum of array from index 5 to 5 -> sum = A[5] = 20

for query 4 -> (2, 3, 4)
calculate sum of array from index 3 to 4 -> sum = A[3]+A[4] = 28

for query 5 -> (2, 3, 3)
calculate sum of array from index 3 to 3 -> sum = A[3] = 12

Hence, answer is 20+28+12 = 60

Sample Input 2
7
3
7
4
2
5
3
7
5
1 0 4
2 0 1
1 3 6
2 3 3
2 0 5

Sample Output 2
111

Sample Output Description 2
Here, n = 7
A = [3, 7, 4, 2, 5, 3, 7]
q = 5
queries = [[1, 0, 4], [2, 0, 1], [1, 3, 6], [2, 3, 3], [2, 0, 5]]

for query 1 -> (1, 0, 4)
Applying the operation on subarray from index 0 to 4, A becomes, A = [3, 6, 9, 12, 15, 3, 7]

for query 2 -> (2, 0, 1)
calculate sum of array from index 0 to 1 -> sum = A[0]+A[1] = 9

for query 3 -> (1, 3, 6)
Applying the operation on subarray from index 3 to 6, A becomes, A = [3, 6, 9, 12, 24, 36, 48]

for query 4 -> (2, 3, 3)
calculate sum of array from index 3 to 3 -> sum = A[3] = 12

for query 5 -> (2, 0, 5)
calculate sum of array from index 0 to 5 -> sum = A[0]+A[1]+A[2]+A[3]+A[4]+A[5] = 90
Hence, answer is 9+12+90 = 111

Sample Input 3
7
1
8
6
10
5
6
9
5
2 0 3
1 2 3
1 0 6
2 1 4
2 6 6

Sample Output 3
46

Sample Output Description 3
Here, n = 7
A = [1, 8, 6, 10, 5, 6, 9]
q = 5
queries = [[2, 0, 3], [1, 2, 3], [1, 0, 6], [2, 1, 4], [2, 6, 6]]

for query 1 -> (2, 0, 3)
calculate sum of array from index 0 to 3 -> sum = A[0]+A[1]+A[2]+A[3] = 25

for query 2 -> (1, 2, 3)
Applying the operation on subarray from index 2 to 3, A becomes, A = [1, 8, 6, 12, 5, 6, 9]

for query 3 -> (1, 0, 6)
Applying the operation on subarray from index 0 to 6, A becomes, A = [1, 2, 3, 4, 5, 6, 7]

for query 4 -> (2, 1, 4)
calculate sum of array from index 1 to 4 -> sum = 14

for query 5 -> (2, 6, 6)
calculate sum of array from index 6 to 6 -> sum = 7

Hence, answer is 25+14+7 = 46

Input Format
The first line contains an integer, n, denoting the number of elements in A.
Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer describing A[i].
The next line contains an integer, q, denoting the number of rows in queries.
Each line i of the q subsequent lines (where 0 ≤ i < q) contains 3 space separated integers each describing the row queries[i].
The 3 space separated integers denote the value of either (1,l,r) or (2,l,r) for the i-th query.

Constraints
1 <= n <= 10^5
1 <= A[i] <= 10^5
1 <= q <= 10^5
0 <= queries[i][j] <= 10^5

Sample Test Cases

Case 1
Input:
7
1
4
5
1
6
7
8
5
1 1 6
1 1 5
2 5 5
2 3 4
2 3 3

Output:
60
Explanation:
Here, n = 7

A = [1, 4, 5, 1, 6, 7, 8]
q = 5
queries = [[1, 1, 6], [1, 1, 5], [2, 5, 5], [2, 3, 4], [2, 3, 3]]

for query 1 -> (1, 1, 6)
Applying the operation on subarray from index 1 to 6, A becomes, A = [1, 4, 8, 12, 16, 20, 24]

for query 2 -> (1, 1, 5)
Applying the operation on subarray from index 1 to 5, A becomes, A = [1, 4, 8, 12, 16, 20, 24]

for query 3 -> (2, 5, 5)
calculate sum of array from index 5 to 5 -> sum = A[5] = 20
for query 4 -> (2, 3, 4)
calculate sum of array from index 3 to 4 -> sum = A[3]+A[4] = 28

for query 5 -> (2, 3, 3)
calculate sum of array from index 3 to 3 -> sum = A[3] = 12

Hence, answer is 20+28+12 = 60

Case 2

Input:

7
3
7
4
2
5
3
7
5
1 0 4
2 0 1
1 3 6
2 3 3
2 0 5

Output:

111

Explanation:

Here, n = 7
A = [3, 7, 4, 2, 5, 3, 7]
q = 5
queries = [[1, 0, 4], [2, 0, 1], [1, 3, 6], [2, 3, 3], [2, 0, 5]]

for query 1 -> (1, 0, 4)
Applying the operation on subarray from index 0 to 4, A becomes, A = [3, 6, 9, 12, 15, 3, 7]

for query 2 -> (2, 0, 1)
calculate sum of array from index 0 to 1 -> sum = A[0]+A[1] = 9

for query 3 -> (1, 3, 6)
Applying the operation on subarray from index 3 to 6, A becomes, A = [3, 6, 9, 12, 24, 36, 48]

for query 4 -> (2, 3, 3)
calculate sum of array from index 3 to 3 -> sum = A[3] = 12

for query 5 -> (2, 0, 5)
calculate sum of array from index 0 to 5 -> sum = A[0]+A[1]+A[2]+A[3]+A[4]+A[5] = 90
Hence, answer is 9+12+90 = 111

Case 3

Input:

7
1
8
6
10
5
6
9
5
2 0 3
1 2 3
1 0 6
2 1 4
2 6 6

Output:

46

Explanation:

Here, n = 7
A = [1, 8, 6, 10, 5, 6, 9]
q = 5
queries = [[2, 0, 3], [1, 2, 3], [1, 0, 6], [2, 1, 4], [2, 6, 6]]

for query 1 -> (2, 0, 3)
calculate sum of array from index 0 to 3 -> sum = A[0]+A[1]+A[2]+A[3] = 25

for query 2 -> (1, 2, 3)
Applying the operation on subarray from index 2 to 3, A becomes, A = [1, 8, 6, 12, 5, 6, 9]

for query 3 -> (1, 0, 6)
Applying the operation on subarray from index 0 to 6, A becomes, A = [1, 2, 3, 4, 5, 6, 7]

for query 4 -> (2, 1, 4)
calculate sum of array from index 1 to 4 -> sum = 14

for query 5 -> (2, 6, 6)
calculate sum of array from index 6 to 6 -> sum = 7

Hence, answer is 25+14+7 = 46

2. EASY

You are given an array A of length N and an integer k.

It is given that a subarray from l to r is considered good, if the number of distinct elements in that subarray doesn't exceed k. Additionally, an empty subarray is also a good subarray and its sum is considered to be zero.

Find the maximum sum of a good subarray.

Sample Input 1
11
2
1
2
2
3
2
3
5
1
2
1
1

Sample Output 1
12

Sample Output Description 1
Here, N = 11, k = 2
A = [1, 2, 2, 3, 2, 3, 5, 1, 2, 1, 1]

We can select the subarray = [2, 2, 3, 2, 3]
It is a good subarray because it contains at most k distinct elements.
Its sum = 2+2+3+2+3 = 12

So, our answer is 12.

Sample Input 2
3
1
-1
-2
-3

Sample Output 2
0
Sample Output Description 2
Here, N = 3, k = 1
A = [-1, -2, -3]

It is optimal to choose empty subarray.

So, our answer is 0.

Sample Input 3
5
5
-1
1
3
2
-1

Sample Output 3
6

Sample Output Description 3
Here, N = 5, k = 5
A = [-1, 1, 3, 2, -1]

It is optimal to choose the subarray = [1, 3, 2].
Its sum = 1+3+2 = 6.
Hence, our answer is 6.

Input Format
The first line contains an integer, N, denoting the number of elements in A.
The next line contains an integer, k, denoting the limit on the distinct elements.
Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing A[i].

Constraints
1 <= N <= 10^5
1 <= k <= n
-10^5 <= A[i] <= 10^5

Sample Test Cases

Case 1
Input:
11
2
1
2
2
3
2
3
5
1
2
1
1

Output:
12

Explanation:
Here, N = 11, k = 2
A = [1, 2, 2, 3, 2, 3, 5, 1, 2, 1, 1]
We can select the subarray = [2, 2, 3, 2, 3]
It is a good subarray because it contains at most k distinct elements.
Its sum = 2+2+3+2+3 = 12
So, our answer is 12.

Case 2

Input:
3
1
-1
-2
-3
Output:
0
Explanation:
Here, N = 3, k = 1
A = [-1, -2, -3]

It is optimal to choose empty subarray.
So, our answer is 0.

Case 3

Input:
5
5
-1
1
3
2
-1
Output:
6
Explanation:
Here, N = 5, k = 5
A = [-1, 1, 3, 2, -1]

It is optimal to choose the subarray = [1, 3, 2].
Its sum = 1+3+2 = 6.
Hence, our answer is 6.

3. EASY

You have an oil tank with a capacity of C litres that can be bought and sold by N people. The people are standing in a queue are served sequentially in the order of array A.

Some of them want to sell a litre of oil and some of them want to buy a litre of oil and A describes this. Here, A[i] = 1 denotes that the person wants to sell a litre of oil and A[i] = -1 denotes that the person wants to buy a litre of oil.

When a person wants to sell a litre of oil but the tank is full, they cannot sell it and become upset. Similarly, when a person wants to buy a litre of oil but the tank is empty, they cannot buy it and become upset. Both these cases cause disturbances.

You can minimize the disturbance by filling the tank initially with a certain X litres of oil.

Find the minimum initial amount of oil X that results in the least number of disturbances.

Input Format
The first line contains an integer, N, denoting the number of elements in A.
The next line contains an integer, C, denoting the capacity of the tank.
Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing A[i].

Constraints
1 <= N <= 10^5
1 <= C <= 10^5
-1 <= A[i] <= 1

Sample Test Cases

Case 1

Input:
3
3
-1
1
1

Output:
1
Explanation:
Given N = 3, C = 3, A = [-1, 1, 1].

To avoid disturbance for Person 1, we need at least 1 liter in the tank initially.

After Person 1 buys 1 liter, the tank will be empty.

Person 2 sells 1 liter, so the tank will have 1 liter.

Person 3 sells another liter, so the tank will have 2 liters.

The minimum initial amount X needed to achieve the least number of disturbances is 1 liter.

Case 2

Input:
3
2
-1
-1
1
Output:
2
Explanation:
Given N = 3, C = 2, A = [-1, -1, 1].

To ensure that there are no disturbances:

We need at least 1 liter for Person 1.

We need an additional 1 liter for Person 2, making the total initial amount of oil X = 2.

Thus, the minimum initial amount of oil X required to achieve the least number of disturbances is 2.

Case 3

Input:
4
3
1
1
1
1
Output:
0
Explanation:
Given N = 4, C = 3, A = [1, 1, 1, 1].

1. Person 1 wants to sell 1 liter of oil (A[1] =1). Initially, the tank is empty, so Person 1 can sell 1 liter. The tank now has 1 liter of oil.
2. Person 2 wants to sell 1 liter of oil (A[2]=1). The tank now has 1 liter, so Person 2 can sell 1 liter. The tank now has 2 liters of oil.
3. Person 3 wants to sell 1 liter of oil (A[3]=1). The tank now has 2 liters, so Person 3 can sell 1 liter. The tank now has 3 liters of oil, which is its full capacity.
4. Person 4 wants to sell 1 liter of oil (A[4]=1). The tank is now full (3 liters), so Person 4 cannot sell 1 liter and will be upset.

Given that the tank capacity C is 3 liters and all actions are selling oil, the only disturbance occurs when Person 4 tries to sell oil to a full tank.

Therefore, no initial amount of oil X will change the disturbance for Person 4, as the tank's capacity is already limiting.

Hence, the minimum initial amount of oil X required to achieve the least number of disturbances remains 0.

4. EASY

General Ali has devised a strategic game to reduce an enemy army of N soldiers to just 1 soldier using a minimal number of moves.

The game allows the following three types of moves:

1. Reduce the enemy army by 1 soldier.
2. Reduce the enemy army by half of its current soldiers, rounding down to the nearest integer.
3. Reduce the enemy army by two-thirds of its current soldiers, rounding down to the nearest integer.

Each move must ensure that the resulting number of soldiers is an integer.

Find the minimum number of moves required to reduce enemy army to just 1 soldier.

Input Format
The first line contains an integer, N, denoting the number of enemy soldiers.

Constraints
1 <= N <= 10^9

Sample Test Cases

Case 1
Input:
5
Output:
3
Explanation:
Given N = 5.

Move 1: Reduce by 1 soldier (5 -> 4)
Move 2: Reduce by half (4 -> 2)
Move 3: Reduce by half (2 -> 1)

Hence, the answer for this case is equal to 3.

Case 2
Input:
1
Output:
0
Explanation:
Given N = 1.

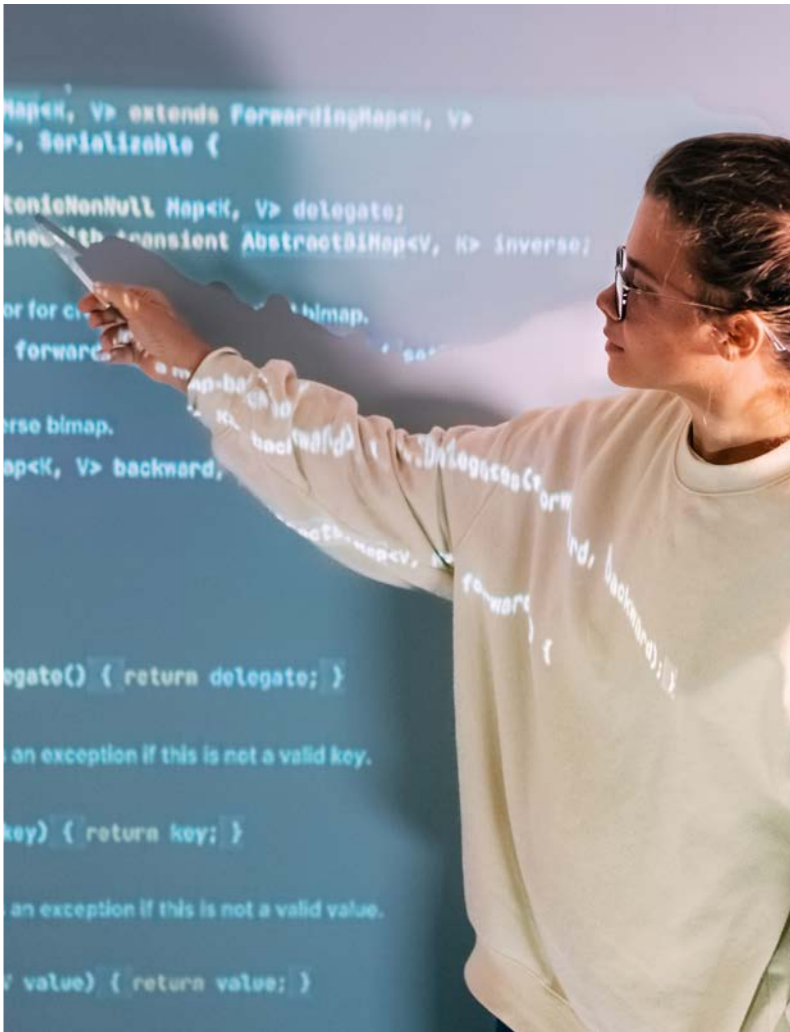
There is only 1 soldier already, so no moves are required to reduce the enemy soldiers to 1.

Therefore, the minimum number of moves needed is 0.

Case 3
Input:
6
Output:
2
Explanation:
Given N = 6.

Move 1: Reduce by half (6 -> 3)
Move 2: Reduce by half (3 -> 1)

Hence, the answer for this case is equal to 2.



5. EASY

General Ali has initiated an invasion in the shape of an N×M grid behind enemy lines given by a 2D array Q.

The grid consists of cells represented by the following characters:

1. '*': Represents a block cell that cannot be visited.
2. 'A': Represents a cell that has been invaded by General Ali's army.
3. 'E': Represents a cell that contains the enemy's army.

General Ali's invasion progresses as follows: At each second, any cell marked 'E' that shares a side with a cell marked 'A' is invaded by General Ali's army.

Find the minimum time it will take for General Ali's army to invade all enemy cells ('E') in the grid. If it's not possible to invade all enemy cells, return –1.

Input Format

The first line contains an integer, N, denoting the number of elements in Q.
The next line contains an integer, M, denoting the given integer.
Each line i of the N subsequent lines (where 0 ≤ i < N) contains a string describing Q[i].

Constraints

1 <= N <= 10^3
1 <= M <= 10^3
1 <= len(Q[i]) <= 10^5

Sample Test Cases

Case 1

Input:
2
2
AE
EE
Output:
2
Explanation:
Given N = 2, M = 2, Q = [["AE"], ["EE"]].

After 1 second the enemy's country will be like Q = [["AA"], ["AE"]].

After 2 seconds the enemy's country will be like Q = [["AA"], ["AA"]].

Hence, the answer for this case is equal to 2.

Case 2

Input:
3
2
AE
*E
EE
Output:
4 **Explanation:**
Given N = 3, M = 2, Q = [["AE"], ["*E"], ["EE"]].

After 1 second the enemy's country will be like Q = [["AA"], ["AE"], ["EE"]].

After 2 seconds the enemy's country will be like Q = [["AA"], ["AA"], ["AE"]].

After 3 seconds the enemy's country will be like Q = [["AA"], ["AA"], ["AA"]]

Hence, the answer for this case is equal to 4.
Case 3
Input:
3
2
AE
**
EE
Output:
-1
Explanation:
Given N = 3, M = 2, Q = [["AE"], ["**"], ["EE"]].

After 1 second the enemy's country will be like Q = [["AA"], ["**"], ["AA"]].

After that General Ali's army can't continue the invasion because of the block cells so the answer is -1.

6. MEDIUM

A company ABC has N employees.

For some reason, the company's building is a bit weird.

- It has one office on each floor and in each office works one employee.
- Each employee i works on the ith floor and has skill Ai.
- Each employee can belong to at most one team.
- Each team should have employees working in consecutive floors from i to j. In other words, the teams should be divided in such a way that no employee of one team can walk into the project space of another team.

ABC uses a metric which is called the expert number which is calculated as the sum of all the absent expert values from each team of employees. The absent expert value of each team is the first skill starting from 0 which is not present in the team.

It is given that a bigger expert number is a better expert number. Hence, you need to divide the employees into teams such that the company's expert number is as large as possible.

Find the maximum expert number that can be obtained.

Sample Input 1

4
0
2
1
1

Sample Output 1

3

Sample Output Description 1

Here, N = 4
A = [0, 2, 1, 1]

We can divide the employees in the following teams -> [0, 2, 1], [1]

expert value of team [0, 2, 1] = 3
expert value of team [1] = 0

So, expert number = 3+0 = 3

Please note that there is no other way to divide the employees such that expert number is greater than 3.

Hence, the maximum value of expert number is 3.

Sample Input 2

5
0
1
2
1
0

Sample Output 2

5

Sample Output Description 2

Here, N = 5
A = [0, 1, 2, 1, 0]

We can divide the employees in the following teams -> [0, 1, 2], [1, 0]

expert value of team [0, 1, 2] = 3
expert value of team [1, 0] = 2

So, expert number = 3+2 = 5

Please note that there is no other way to divide the employees such that expert number is greater than 5.

Hence, the maximum value of expert number is 5.

Sample Input 3

10
0
1
0
1

1
0
3
2
1
0

Sample Output 3
10

Sample Output Description 3
Here, N = 10
A = [0, 1, 0, 1, 1, 0, 3, 2, 1, 0]

We can divide the employees in the following teams
-> [0, 1], [0, 1], [1, 0, 3, 2], [1, 0]

expert value of team [0, 1] = 2
expert value of team [1, 0, 3, 2] = 4

So, expert number = 2 + 2 + 4 + 2 = 10

Please note that there is no other way to divide the employees such that expert number is greater than 10.

Hence, the maximum value of expert number is 10.

Input Format
The first line contains an integer, N, denoting the number of elements in A.
Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing A[i].

Constraints
1 <= N <= 10^5
1 <= A[i] <= 10^3
Sample Test Cases

Case 1
Input:
4
0
2
1
1
Output:
3
Explanation:
Here, N = 4
A = [0, 2, 1, 1]
We can divide the employees in the following teams
-> [0, 2, 1], [1]

expert value of team [0, 2, 1] = 3
expert value of team [1] = 0

So, expert number = 3+0 = 3

Please note that there is no other way to divide the employees such that expert number is greater than 3.

Hence, the maximum value of expert number is 3.

Case 2
Input:
5
0
1
2
1
0
Output:
5
Explanation:
Here, N = 5
A = [0, 1, 2, 1, 0]

We can divide the employees in the following teams -> [0, 1, 2], [1, 0]

expert value of team [0, 1, 2] = 3
expert value of team [1, 0] = 2

So, expert number = 3+2 = 5

Please note that there is no other way to divide the employees such that expert number is greater than 5.

Hence, the maximum value of expert number is 5.

Case 3

Input:
10
0
1
0
1
1
0
3
2
1
0
Output:
10
Explanation:
Here, N = 10
A = [0, 1, 0, 1, 1, 0, 3, 2, 1, 0]

We can divide the employees in the following teams -> [0, 1], [0, 1], [1, 0, 3, 2], [1, 0]

expert value of team [0, 1] = 2
expert value of team [1, 0, 3, 2] = 4

So, expert number = 2 + 2 + 4 + 2 = 10

Please note that there is no other way to divide the employees such that expert number is greater than 10.

Hence, the maximum value of expert number is 10.

7. MEDIUM

You are given a graph with n nodes. Initially, the graph is disconnected, meaning it contains zero edges.

Each node has a value written on it such that the ith node has a value i.

We say that a range [l, r] is covered in a set s, if for each i from l to r, i appears in s.

Now, let's define beauty(s), as the minimum number of covered ranges, such that each element of s belongs to at least one of these ranges. For example: beauty([1, 2, 4, 5, 8, 11]) = 4 (covered ranges are [1, 2], [4, 5], [8] and [11]).

You have to process q queries that are either of the following types:

- Type 1 - (1, i, j): Add an edge between i and j.
- Type 2 - (2, u, 0): Find the number of covered ranges in the connected component of u.

Find the sum of answers to all type 2 queries.

Sample Input 1

2
1
3
2 1 0

Sample Output 1

1

Sample Output Description 1

Here, n = 2, q = 1, t = 3
queries = [[2, 1, 0]]

for query1 -> As there are no edges in the graph, the connected component of 1 contain only node 1. only one covered range required i.e. [1, 1]
Hence, answer for this query is 1.

So, the answer is 1.

Sample Input 2

2
3
3
2 1 0

1 1 2
2 1 0

Sample Output 2
2

Sample Output Description 2
Here, n = 2, q = 3, t = 3
queries = [[2, 1, 0], [1, 1, 2], [2, 1, 0]]

for query1 -> (2, 1, 0)
connected component of 1 = {1}
only one covered range required i.e. [1, 1]
Hence, answer for this query is 1.

for query2 -> (1, 1, 2)
Add an edge between 1 and 2.
for query3 -> (2, 1, 0)
connected component of 1 = {1, 2}
only one covered range required i.e. [1, 2]
Hence, answer for this query is 1.
So, the answer is 1 + 1 = 2.

Sample Input 3
10
3
3
1 1 4
2 1 0
2 4 0

Sample Output 3
4

Sample Output Description 3
Here, n = 10, q = 3, t = 3
queries = [[1, 1, 4], [2, 1, 0], [2, 4, 0]]

for query1 -> (1, 1, 4)
Add an edge between 1 and 4

for query2 -> (2, 1, 0)
connected component of 1 = {1, 4}
two covered ranges are required i.e. [1, 1] and [4, 4]
Hence, answer for this query is 2.

for query3 -> (2, 4, 0)
connected component of 4 = {1, 4}
two covered ranges are required i.e. [1, 1] and [4, 4]
Hence, answer for this query is 2.
So, the answer is 2 + 2 = 4.

Input Format
The first line contains an integer, n, denoting the size of the graph.
The next line contains an integer, q, denoting the number of rows in queries.
The next line contains an integer, t, denoting the number of columns in queries.
Each line i of the q subsequent lines (where 0 ≤ i < q) contains t space separated integers each describing the row queries[i].

Constraints
1 <= n <= 10^5
1 <= q <= 10^5
3 <= t <= 3
0 <= queries[i][j] <= n

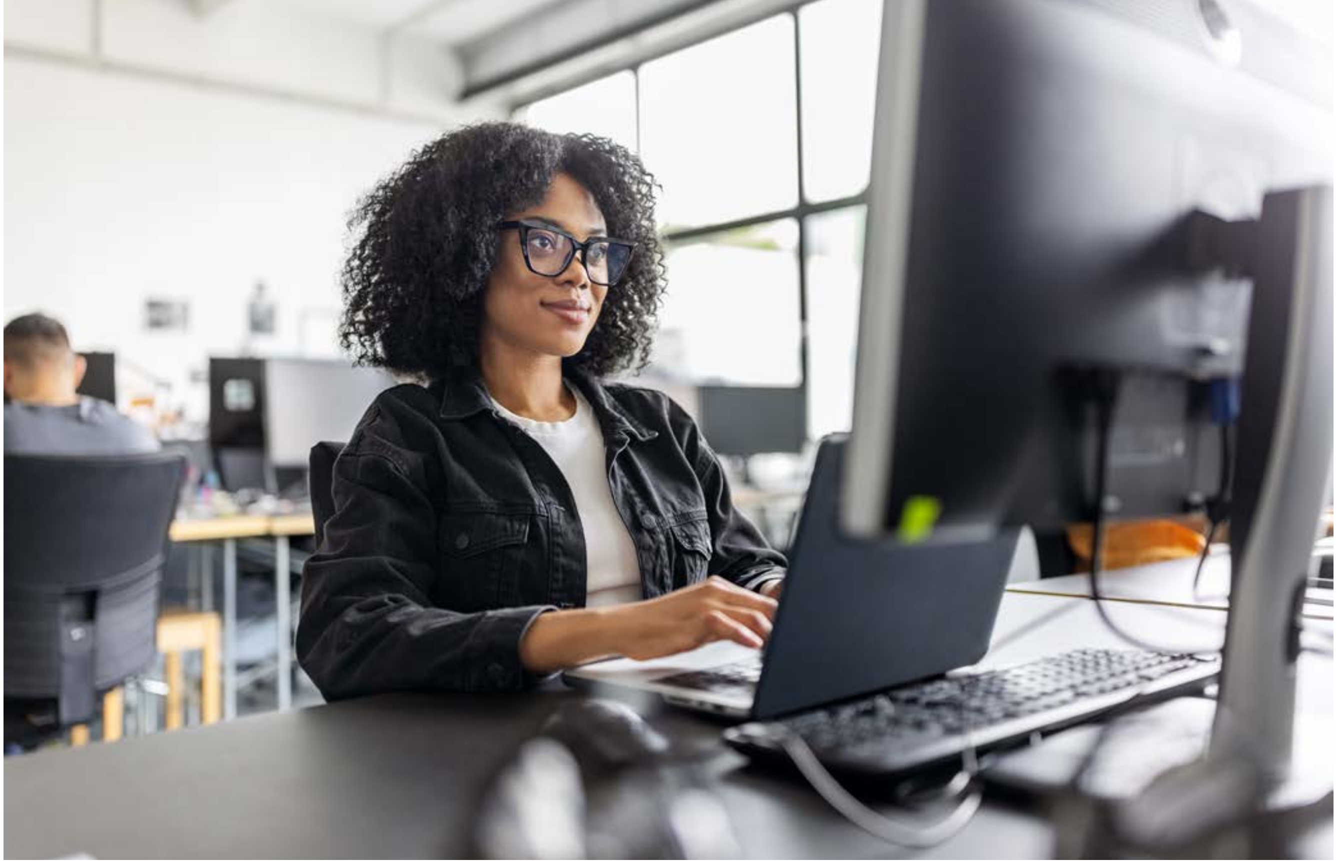
Sample Test Cases

Case 1
Input:
2
1
3
2 1 0
Output:
1
Explanation:
Here, n = 2, q = 1, t = 3
queries = [[2, 1, 0]]

for query1 -> As there are no edges in the graph, the connected component of 1 contain only node 1.
only one covered range required i.e. [1, 1]
Hence, answer for this query is 1.

So, the answer is 1.

Case 2
Input:
2
3
3
2 1 0
1 1 2
2 1 0



Output:
2
Explanation:
Here, n = 2, q = 3, t = 3
queries = [[2, 1, 0], [1, 1, 2], [2, 1, 0]]

for query1 -> (2, 1, 0)
connected component of 1 = {1}
only one covered range required i.e. [1, 1]
Hence, answer for this query is 1.

for query2 -> (1, 1, 2)
Add an edge between 1 and 2.

for query3 -> (2, 1, 0)
connected component of 1 = {1, 2}
only one covered range required i.e. [1, 2]
Hence, answer for this query is 1.
So, the answer is 1 + 1 = 2.

Case 3

Input:
10
3
3
1 1 4
2 1 0
2 4 0
Output:
4
Explanation:
Here, n = 10, q = 3, t = 3
queries = [[1, 1, 4], [2, 1, 0], [2, 4, 0]]

for query1 -> (1, 1, 4)
Add an edge between 1 and 4

for query2 -> (2, 1, 0)
connected component of 1 = {1, 4}
two covered ranges are required i.e. [1, 1] and [4, 4]
Hence, answer for this query is 2.

for query3 -> (2, 4, 0)
connected component of 4 = {1, 4}
two covered ranges are required i.e. [1, 1] and [4, 4]
Hence, answer for this query is 2.
So, the answer is 2 + 2 = 4

8. MEDIUM

A group of N people are seated around a circular table to play a game.

The game involves jumping from one chair to another. Each person sitting on chair i can jump A[i] chairs to either the right or left in one jump where 0 < i < N+1.

Bob, sitting on chair X, needs to reach chair Y, where the escape door is located.

Find the minimum number of jumps required to reach chair Y from chair X. If this is impossible using the given jump distances, then return -1.

Input Format
The first line contains an integer, N, denoting the number of people playing the game.
The next line contains an integer, X, denoting the chair on which Bob is seated.
The next line contains an integer, Y, denoting the chair which Bob wants to reach.
Each line i of the N subsequent lines (where 1 ≤ i ≤ N) contains an integer describing A[i] is the number of chairs the person sitting in chair number i can jump either right or left.

Constraints
1 <= N <= 10^5
1 <= X <= N
1 <= Y <= N
1 <= A[i] <= 10^5

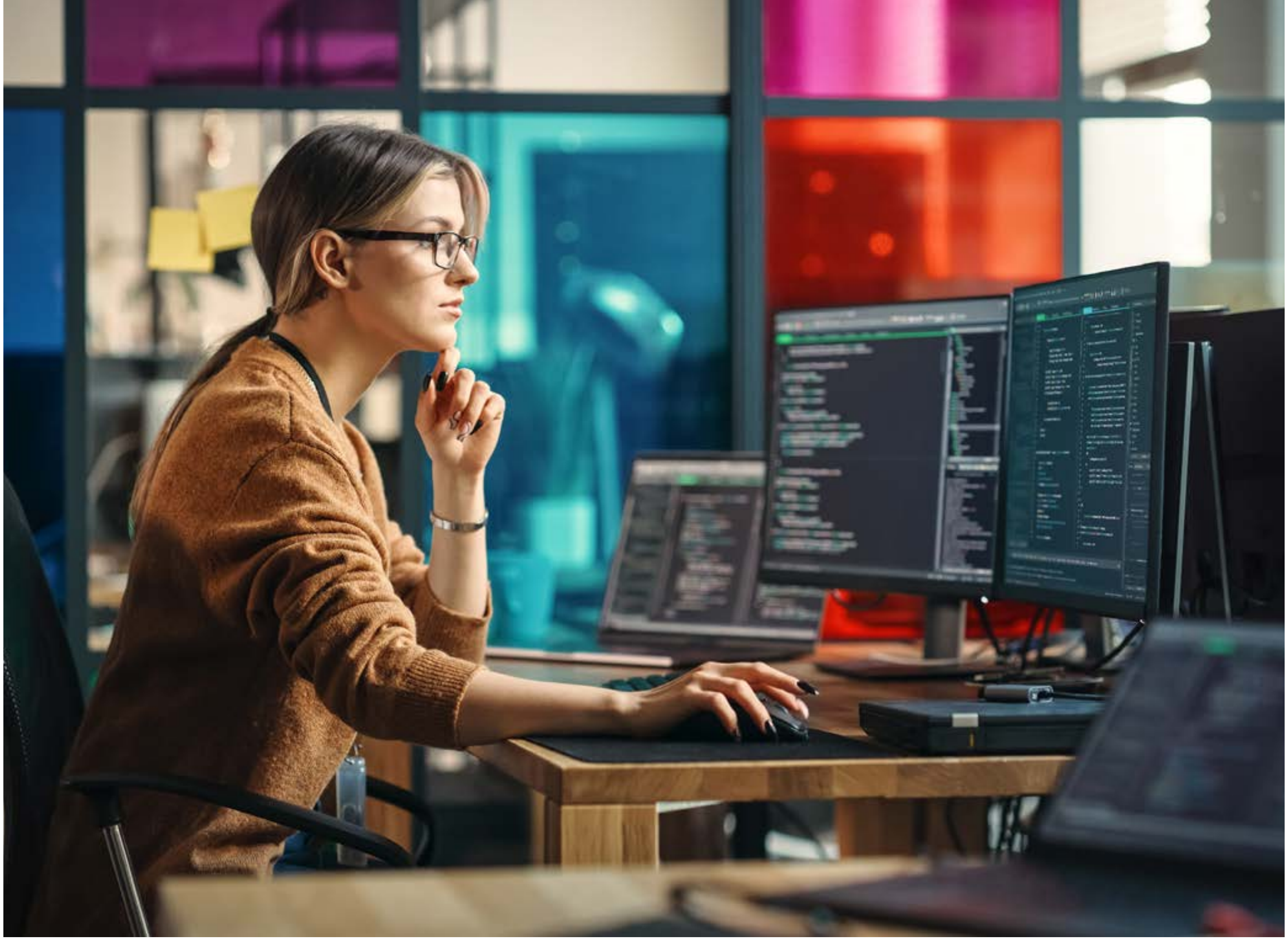
Sample Test Cases

Case 1
Input:
5
5
1
1
2
3
2
4
Output:
1

Case 2
Input:
5
2
4
5
4
3
2
1

Output:
3

Case 3
Input:
6
2
3
2
2
2
2
2
2
Output:
-1



9. MEDIUM

You are given two boxes where one contains an infinite number of blue balls and the other contains an infinite number of red balls. We can construct a chain of balls by choosing one of the boxes and taking out a ball from it and inserting it at the end of the chain.

A chain is called good if after each time we insert a new ball, the number of blue balls doesn't exceed the number of red balls by more than K. In other words, if after adding the ith ball we have B[i] blue balls and R[i] red balls then it must satisfy B[i] ≤ R[i] + K.

A chain is called great if it is a good chain and if we take the reversed chain the following conditions satisfy:

- Every blue ball in the reversed chain is matched with a red ball in the original chain.
- Every red ball in the reversed chain is matched with a blue ball in the original chain.

You are given two integers B and R. Let the probability of choosing the box with the blue balls be (B / 106) and (R / 106) be the probability of choosing the box with red balls.

Find the probability of creating a great chain of length N. Since the answer can be large, return it modulo 109 + 7.

Input Format
The first line contains an integer, N, denoting the length of the chain.
The next line contains an integer, K, denoting the value K in B[i] ≤ R[i] + K.
The next line contains an integer, B, denoting the probability of choosing the box with blue balls.
The next line contains an integer, R, denoting the probability of choosing the box with red balls.

Constraints
1 <= N <= 10^9
1 <= K <= 100
1 <= B <= 10^6
1 <= R <= 10^6

Sample Test Cases

Case 1

Input:
1
1
199252
470888
Output:
542964004
Explanation:
Given N = 1, M = 1, B = 199252, R = 470888.

There is only one great chain: B.

Hence, the output for this case is equal to 542964004.

Case 2

Input:
2
1
748096
475634
Output:
170882874
Explanation:
Given N = 2, M = 1, B = 748096, R = 475634.

There are two great chains: BR, BB.
Hence, the output for this case is equal to 170882874.

Case 3

Input:
4
3
813081
102149
Output:
6235092
Explanation:
Given N = 4, M = 3, B = 813081, R = 102149.

The output for this case is equal to 6235092.

10. MEDIUM

You are given an array A of size N.

You can partition A into multiple subarrays such that each element belongs to exactly one subarray and each subarray has a length of at least K.

The beauty of a subarray is the maximum bitwise XOR of the values of a subset in that subarray. The amazingness of a partitioned array is the sum of beauties of its subarrays.

Find the maximum possible amazingness of A.

Note:
- A subarray is a contiguous part of the array.

Input Format
The first line contains an integer, N, denoting the number of elements in A.
The next line contains an integer, K, denoting the given integer.
Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing A[i].

Constraints
1 <= N <= 10^5
1 <= K <= 10^5
1 <= A[i] <= 10^5

Sample Test Cases

Case 1

Input:
2
2
2
1
Output:
3
Explanation:
Given N = 2, K = 2, A = [2, 1].

We take the entire A as one subarray as [2, 1] with maximum amazingness equal to 3.
Hence, the answer for this case is equal to 3.

Case 2

Input:
4
1
1
5
3
3
Output:
12
Explanation:
Given N = 4, K = 1, A = [1, 5, 3, 3].

We can take 4 subarrays from A as [1], [5], [3], [3] whose maximum amazingness is equal to 1 + 5 + 3 + 3 = 12.
Hence, the answer for this case is equal to 12.

Case 3

Input:
7
1
16
3
3
5
19
19
5
Output:
70
Explanation:
Given N = 7, K = 1, A = [16, 3, 3, 5, 19, 19, 5].

The maximum possible amazingness for this case is equal to 70.

11. HARD

You are given a permutation p of length n and an integer m. You now need to construct a directed graph from the given permutation.

It is given that an edge exists between i and j if ai < aj and abs(i - j) <= k, where abs(x) is the absolute value of x.

Find the minimum value of k such that the longest path of the resulting graph is greater than or equal to m.

Notes:

- The length of the path is equal to the number of nodes in that path.

Sample Input 1

5
2
1
3
2
5
4

Sample Output 1

1

Sample Output Description 1

Here, n = 5, m = 2
p = [1, 3, 2, 5, 4]

If k >= 1 then, directed edge 1 -> 2 exists, so, we can take the path 1 -> 2.
Hence, the minimum value of k required is 1.

So, answer is 1.

Sample Input 2

5
3
1
3
2
5
4

Sample Output 2
2

Sample Output Description 2
Here, n = 5, m = 3
p = [1 ,3, 2, 5, 4]

If k >= 2, then we can take the path 1 -> 2 -> 3.
Hence, the minimum value of k required is 2.

So, the answer is 2.

Sample Input 3
5
1
1
2
3
4
5

Sample Output 3
0

Sample Output Description 3
Here, n = 5, m = 1
p = [1, 2, 3, 4, 5]

Any node (without any edges) has a path of length 1, so the minimum value k is 0.
So, answer is 0.

Input Format
The first line contains an integer, n, denoting the number of elements in p.
The next line contains an integer, m, denoting the length of the path.
Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer describing p[i].

Constraints
1 <= n <= 10^5
1 <= m <= n
1 <= p[i] <= n

Sample Test Cases

Case 1
Input:
5
2
1
3
2
5
4
Output:
1

Explanation:
Here, n = 5, m = 2
p = [1, 3, 2, 5, 4]

If k >= 1 then, directed edge 1 -> 2 exists, so, we can take the path 1 -> 2.
Hence, the minimum value of k required is 1.
So, answer is 1.

Case 2

Input:
5
3
1
3
2
5
4
Output:
2
Explanation:
Here, n = 5, m = 3
p = [1 ,3, 2, 5, 4]

If k >= 2, then we can take the path 1 -> 2 -> 3.
Hence, the minimum value of k required is 2.
So, the answer is 2.

Case 3

Input:
5
1
1
2
3
4
5
Output:
0
Explanation:
Here, n = 5, m = 1
p = [1, 2, 3, 4, 5]

Any node (without any edges) has a path of length 1, so the minimum value k is 0.
So, answer is 0.

12. HARD

Bob is playing a game called "Some Help".

In this game, there are N soldiers, where N is an even number. There are also N treasure chests, each with a bonus value given by the array Bonus. Here, Bonus[i] denotes the bonus value for the ith chest.

The power of each soldier is described by an array A of size N. For each i, the power of the ith soldier is an integer between 1 and N/2, and each number between 1 and N/2 occurs exactly twice in A.

The game has N rounds and each round proceeds as follows:

1. For each ith player, Bob finds the first player on their right whose power is a multiple of the power of the ith player. Let's call this player R.
2. If no such player R is found, Bob does nothing.
3. If such a player R is found, Bob can choose a chest in the range [i, R] that gives the maximum bonus. Let's call the bonus of this chest as X.
4. The total XP is initially zero and for each round it is increased by X.

Find the total XP that Bob can obtain from all N rounds.

Note:

- Each treasure chest can be used any number of times.

Input Format
The first line contains an integer, N, denoting the number of elements in A.
Each line i of the N subsequent lines (where 1 ≤ i ≤ N) contains an integer describing A[i].
Each line i of the N subsequent lines (where 1 ≤ i ≤ N) contains an integer describing Bonus[i].

Constraints
1 <= N <= 10^5
1 <= A[i] <= 10^5
1 <= Bonus[i] <= 10^5

Sample Test Cases

Case 1

Input:
4
1
1
2
2
4
8
2
1
Output:
18
Explanation:
Given N = 4, A = [1, 1, 2, 2], Bonus = [4, 8, 2, 1].

1. For the first soldier (power 1):
   The next soldier to the right with a power that is a multiple of 1 is the second soldier (power 1).
   The maximum bonus in the range [1st soldier, 2nd soldier] is 8.
   Add 8 to the total XP.
2. For the second soldier (power 1):
   The next soldier to the right with a power that is a multiple of 1 is the third soldier (power 2).
   The maximum bonus in the range [2nd soldier, 3rd soldier] is 8.
   Add 8 to the total XP.
3. For the third soldier (power 2):
   The next soldier to the right with a power that is a multiple of 2 is the fourth soldier (power 2).
   The maximum bonus in the range [3rd soldier, 4th soldier] is 2.
   Add 2 to the total XP.
4. The fourth soldier (power 2) has no subsequent soldiers to the right with a power that is a multiple of 2, so no bonus is added.
   Total XP = 8 (from soldier 1) + 8 (from soldier 2) + 2 (from soldier 3) = 18
   Therefore, the total XP that Bob can get from N rounds is 18.

Case 2

Input:
6
1
2
3
1
2
3
4
2
1
4
5
9
Output:
23
Explanation:
Given N = 6, A = [1, 2, 3, 1, 2, 3], Bonus = [4, 2, 1, 4, 5, 9].

1. For soldier 1 (power 1):
The first soldier to the right whose power is a multiple of 1 is soldier 2 (power 2).
The maximum bonus in the range [1, 2] is max (4, 2) = 4. Add 4 to the total XP.

2. For soldier 2 (power 2):
The first soldier to the right whose power is a multiple of 2 is soldier 5 (power 2).
The maximum bonus in the range [2, 5] is max (2, 1, 4, 5) = 5. Add 5 to the total XP.

3. For soldier 3 (power 3):
The first soldier to the right whose power is a multiple of 3 is soldier 6 (power 3).
The maximum bonus in the range [3, 6] is max(1, 4, 5, 9) = 9. Add 9 to the total XP.

4. For soldier 4 (power 1):
There are no soldiers to the right whose power is a multiple of 1. No bonus is added.

5. For soldier 5 (power 2):
There are no soldiers to the right whose power is a multiple of 2. No bonus is added.

6. For soldier 6 (power 3):
There are no soldiers to the right whose power is a

multiple of 3. No bonus is added.

Total XP = 4+5+9+5 = 23.
Case 3
Input:
4
1
1
2
2
16
8
4
2

Output:
28
Explanation:
Given N = 4, A = [1, 1, 2, 2], Bonus = [16, 8, 4, 2].

1. For soldier 1 (power 1):
The next soldier to the right whose power is a multiple of 1 is the second soldier (power 1). The maximum bonus in the range [1, 2] is max(16,8)=16. Add 16 to the total XP.

2. For soldier 2 (power 1):
The next soldier to the right whose power is a multiple of 1 is the third soldier (power 2). The maximum bonus in the range [2, 3] is max(8,4)=8. Add 8 to the total XP.

3. For soldier 3 (power 2):
The next soldier to the right whose power is a multiple of 2 is the fourth soldier (power 2). The maximum bonus in the range [3, 4] is max(4,2)=4. Add 4 to the total XP.

4. For soldier 4 (power 2):
There are no soldiers to the right whose power is a multiple of 2. No bonus is added.

Total XP = 16 + 8 + 4 = 28.

13. HARD

You are given an array A of length N.

You have two functions defined as follows:

- frequency(left, right, value): Returns the number of elements in the range [left, right] that are equal to value.
- distinct(left, right): Returns the number of distinct elements in the range [left, right].

Find the number of pairs (i, j) that satisfy the following condition:

- frequency(1 , i , A[i]) + frequency(j , N , A[j]) ≤ .[distinct(1 , i) / 2] + [distinct(j , N) / 2].

Since the answer can be very large, return it modulo 109+7.

Note:

- Here X represents the greatest integer less than or equal to X.
- 1 ≤ i < j ≤ N

Input Format

The first line contains an integer, N, denoting the number of elements in A.
Each line i of the N subsequent lines (where 0 ≤ i < N) contains an integer describing A[i].

Constraints

1 <= N <= 10^5
1 <= A[i] <= 10^9

Sample Test Cases

Case 1

Input:
5
2
2
3
1
5
Output:
2
Explanation:
Given N = 5, A = [2, 2, 3, 1, 5].

We can select two pairs (1, 2) and (3,4) which satisfy the given conditions.

Hence, the answer for this case is equal to 2.

Case 2

Input:
5
5
5
5
5
5
Output:
0
Explanation:
Given N = 5, A = [2, 2, 3, 1, 5].

There exists no pair which satisfy the given conditions.
Hence, the answer for this case is equal to 0.

Case 3

Input:
5
1
2
3
4
5
Output:
5
Explanation:
Given N = 5, A = [1, 2, 3, 4, 5].

There are 5 pairs which satisfy the following conditions are (1, 2), (1, 3), (1, 4), (1, 5), (2, 3).

Hence, the answer for this case is equal to 5.

14. HARD

You are given a tree which consists of N nodes. You also will be given two distinct nodes A and B.

You are also given an array P where the parent of node U is given by P[U].

We define beauty of a node (U, V) as the number of nodes that belong to both:

- The subtree of U when the tree is rooted at A.
- The subtree of U when the tree is rooted at B.

You are given a 2D array Queries which consists of Q queries. For each query you are given two nodes U and V (U can be equal to V) and the answer to the query is the beauty of (U, V).

Let K be equal to the answer to the last query (initially 0), then U and V changes as follows after each query:

- U = (U + K) mod N + 1.
- V = (V + K) mod N + 1.

Find the sum of answer to all queries. Since the answer can be large, return it modulo 109+7.

Input Format

The first line contains an integer, N, denoting the number of elements in P.
The next line contains an integer, A, denoting the given distinct node
The next line contains an integer, B, denoting the given distinct node.
Each line i of the N subsequent lines (where 1 ≤ i ≤ N) contains an integer describing P[i].
The next line contains an integer, Q, denoting the number of rows in Queries.
The next line contains an integer, Col, denoting the number of columns in Queries.
Each line i of the Q subsequent lines (where 0 ≤ i < Q) contains Col space separated integers each describing the row Queries[i].

Constraints

1 <= N <= 10^5
1 <= A <= N
1 <= B <= N
0 <= P[i] <= N
1 <= Q <= 10^5
2 <= Col <= 2
1 <= Queries[i][j] <= N

Sample Test Cases

Case 1

Input:
2
1
2
0
1
1
2
1 2
Output:
0
Explanation:
Given N = 2, A = 1, B = 2, P = [0, 1], Q = 1, Col = 2, Q = [[1, 2]].

Initialize K = 0.

For the query (1, 2), calculate new values of U and V:
U = (1 + K) % N + 1 = (1 + 0) % 2 + 1 = 2
V = (2 + K) % N + 1 = (2 + 0) % 2 + 1 = 1

The transformed query becomes (2, 1).

For the transformed query (2, 1), which in 0-based indexing is (1, 0):

The subtree of node 1 (when rooted at node 0) includes node 1.

The subtree of node 0 (when rooted at node 1) includes node 0.

There is no intersection between the subtrees of node 1 (rooted at node 0) and node 0 (rooted at node 1).

Therefore, the sum of answers to all queries is 0.

Case 2

Input:

2
1
2
0
1
2
2
1 1
1 2

Output:

3

Explanation:

Given N = 2, A = 1, B = 2, P = [0, 1], Q = 2, Col = 2, Q = [[1, 1], [1, 2]].

1. First Query (1, 1):
Calculate new values of U and V:

U = (1 + K) % N + 1 = (1 + 0) % 2 + 1 = 2
V = (1 + K) % N + 1 = (1 + 0) % 2 + 1 = 2

The transformed query becomes (2, 2).

Calculate the beauty of (2, 2):
Since U and V are the same, the beauty is the size of the subtree of node 2 when rooted at node 0 (A):

Subtree size of node 1 (0-based index of node 2) when rooted at 0: 1 (only node 1).

Update K as:
K = Result = 1.

Second Query (1, 2):
Calculate new values of U and V:

U = (1 + K) % N + 1 = (1 + 1) % 2 + 1 = 1
V = (2 + K) % N + 1 = (2 + 1) % 2 + 1 = 2

The transformed query becomes (1, 2).

Calculate the beauty of (1, 2):
Subtree of node 0 (when rooted at node 0) includes nodes 0 and 1.

Subtree of node 1 (when rooted at node 1) includes only node 1.

Intersection of subtrees rooted at nodes 0 and 1 is just node 1.

Therefore, the beauty is 2.

Hence, the sum of answers to all queries is equal to 3.

Case 3

Input:

4
1
2
0
1
1
3
4
2
3 3
1 3
4 4
1 2

Output:

6

Explanation:

Given N = 2, A = 1, B = 2, P = [0, 1], Q = 2, Col = 2, Q = [[1, 1], [1, 2]].

The answers for the 1st, 2nd, 3rd, and 4th query's are equal to 1, 2, 2, and 1 respectively.

Hence, the sum of answers to all queries is equal to 6.

For more information, contact askus@infosys.com

Infosys®
Navigate your next