CopyCopyCopyCopyCopy

locked

Ehab has an array **a** of length **n**. He has just enough free time to make a new array consisting of **n** copies of the old array, written back-to-back. What will be the length of the new array's longest increasing subsequence?

A sequence **a** is a subsequence of an array **b** if **a** can be obtained from **b** by deletion of several (possibly, zero or all) elements. The longest increasing subsequence of an array is the longest subsequence such that its elements are ordered in strictly increasing order.

Input Format

The first line contains an integer **t** — the number of test cases you need to solve. The description of the test cases follows.

The first line of each test case contains an integer **n** ($1 \le n \le 10^5$) — the number of elements in the array **a**. The second line contains **n** space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array **a**

Constraints

$1 \le n \le 10^5$
$1 \le a_i \le 10^9$

Output Format

For each testcase, output the length of the longest increasing subsequence of **a** if you concatenate it to itself **n** times.

Sample Input 0

```
2
3
3 2 1
6
3 1 4 1 5 9
```

Sample Output 0

```
3
5
```

Explanation 0

In the first sample, the new array is **[3,2,1,3,2,1,3,2,1]**. The longest increasing subsequence is marked in bold.

In the second sample, the longest increasing subsequence will be **[1,3,4,5,9]**.

Submissions: 0
Max Score: 0
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++

```
// Sample Input 0

// 2
// 3
// 3 2 1
// 6
// 3 1 4 1 5 9
// Sample Output 0

// 3
// 5




#include<iostream>
#include<bits/stdc++.h>
using namespace std;


int main() {
    long t;
    cin>>t;
    while(t-->0)
    {
        long n;
        cin>>n;
     unordered_set<long> s;
        for(long p=0;p<n;p++)
```

```
        {
            long x;
            cin>>x;
            s.insert(x);
        }
        cout<<s.size()<<"\n";
    }
    return 0;
}
```

Line: 1 Col: 4

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
2
3
3 2 1
6
3 1 4 1 5 9
```

Your Output (stdout)

```
3
5
```

Expected Output

```
3
5
```