Gena Playing Hanoi

by nikasvanidze

The Tower of Hanoi is a famous game consisting of 3 rods and a number of discs of incrementally different diameters. The puzzle starts with the discs neatly stacked on one rod, ordered by ascending size with the smallest disc at the top. The game's objective is to move the entire stack to another rod, obeying the following rules:

1. Only one disc can be moved at a time.
2. Each move consists of taking the topmost disc from a stack and moving it to the top of another stack.
3. No disc may be placed on top of a *smaller* disc.

Gena has a modified version of the Tower of Hanoi. His *Hanoi* has 4 rods and N discs ordered by ascending size. He made a few moves (following the rules above), but stopped and lost his place. He wants to restore the tower to its original state by making valid moves. Given the state of Gena's *Hanoi*, help him calculate the minimum number of moves needed to restore the tower to its original state.

Note: Gena's rods are numbered from 1 to 4. All discs are initially located on rod 1.

Input Format

The first line contains a single integer, N, denoting the number of discs.
The second line contains N space-separated integers, where the i^{th} integer is the index of the rod where the disk with diameter i is located.

Constraints

- $1 \leq N \leq 10$

Output Format

Print the minimum number of moves Gena must make to restore the tower to its initial, ordered state on the first rod.

Sample Input

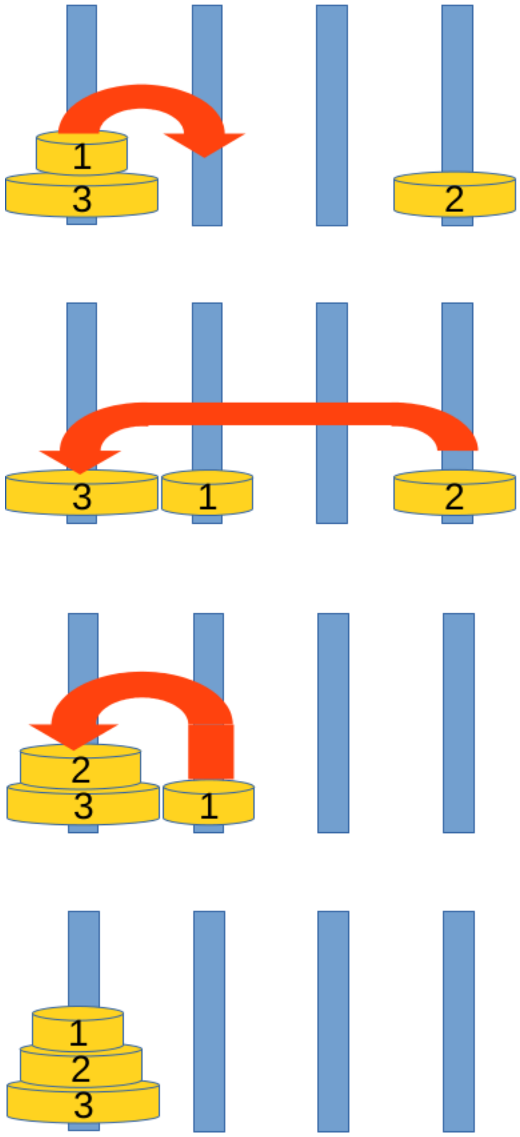
```
3
1 4 1
```

Sample Output

```
3
```

Explanation

3 moves are enough to build the tower. Here is one possible solution:



Submissions:602

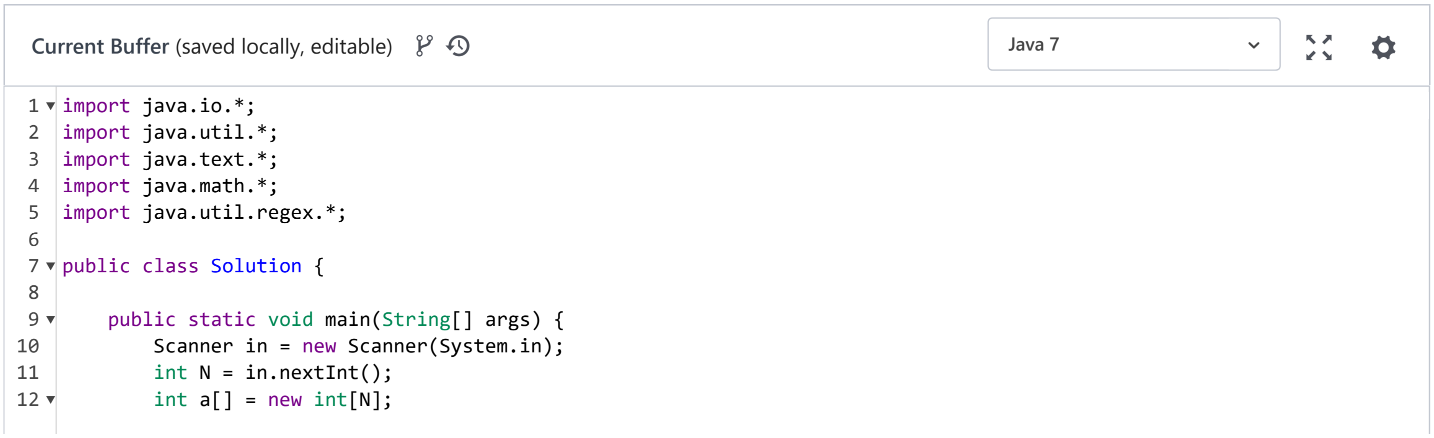
Max Score:50

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

More


Current Buffer (saved locally, editable)

Java 7

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int N = in.nextInt();
        int a[] = new int[N];
        for(int a_i=0; a_i < N; a_i++){
            a[a_i] = in.nextInt();
        }
    }
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input

Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature