Dashboard > Algorithms > Dynamic Programming > Count Scorecards

Badge Progress (Details)

Points: 126   Rank: 270332

Count Scorecards

H by HackerRank

| Problem | Submissions | Leaderboard | Discussions |

In a tournament, n players play against each other exactly once. Each game results in exactly one player winning. There are no ties. You have been given a scorecard containing the scores of each player at the end of the tournament. The score of a player is the total number of games the player won in the tournament. However, the scores of some players might have been erased from the scorecard. How many possible scorecards are consistent with the input scorecard?

Input Format

The first line contains a single integer t denoting the number of test cases. t test cases follow.

The first line of each test case contains a single integer n. The second line contains n space-separated integers $s_1, s_2, \ldots, s_n$. s_i denotes the score of the ith player. If the score of the ith player has been erased, it is represented by -1.

Constraints

- $1 \le t \le 20$
- $1 \le n \le 40$
- $-1 \le s_i < n$

Output Format

For each test case, output a single line containing the answer for that test case modulo $10^9 + 7$.

Sample Input 0

```
5
3
-1 -1 2
3
-1 -1 -1
4
0 1 2 3
2
1 1
4
-1 -1 -1 2
```

Sample Output 0

```
2
7
1
0
12
```

Explanation 0

For the first case, there are 2 scorecards possible: (0,1,2) or (1,0,2).
For the second case, the valid scorecards are (1,1,1), (0,1,2), (0,2,1), (1,0,2), (1,2,0), (2,0,1), (2,1,0).

For the third case, the only valid scorecard is (0,1,2,3).

For the fourth case, there is no valid scorecard. It is not possible for both players to have score of 1.

For the fifth case, 6-variations of {(3,1,0)[2]}, and 3 variations each of {(2,2,0)[2]} and {(2,1,1)[2]}.

Submissions:583

Max Score:100
Difficulty: Expert

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)   Java 7 ⌄

```java
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }
```

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