Find the Derangement of An Array

In combinatorial mathematics, a derangement is a permutation of the elements of a set, such that no element appears in its original position.

There's originally an array consisting of `n` integers from 1 to `n` in ascending order, you need to find the number of derangement it can generate.

Also, since the answer may be very large, you should return the output mod $10^9 + 7$.

Example 1:

```
Input: 3
Output: 2
Explanation: The original array is [1,2,3]. The two derangements are [2,3,1] and [3,1,2].
```

Note:

`n` is in the range of [1, 10^6].

Solution 1

Details can be found from wiki:
https://en.wikipedia.org/wiki/Derangement#Counting_derangements

```
    private static final int M = 1000000007;
    public int findDerangement(int n) {
        long ans = 1;
        for (int i = 1; i <= n; i++)
            ans = (i * ans % M + (i % 2 == 0 ? 1 : -1)) % M;
        return (int) ans;
    }
```

written by MichaelPhelps original link here

Solution 2

The Staggered formula is **D(n) = (n-1) [D(n-2) + D(n-1)]**:

For the k th element, it has k-1 positions and there are two possibilities for its position

- 1.It's not in the first element, so it's going to be the same thing as D(n - 1)
- 2.It's in the position of the first element,so there are two elements in the deranged position.
  So it's going to be the same thing as D(n - 2)

so *res = ((i-1)(dn1+dn2))%1000000007;**
why we use long not int:
a(11) = 14684570
a(12) = 176214841
*a(13) = 12 * (a(12) + a(11)) = 2290792932 > Integer.MAX_VALUE*

```
public int findDerangement(int n) {
        long dn2 = 0, dn1 = 1;
        long res = n==1 ? 0 : 1;
        for (int i = 3; i <= n; i++){
            res = ((i-1) * (dn1+dn2))%1000000007;
            dn2 = dn1;
            dn1 = res;
        }
        return (int) res;
    }
```

written by LeiYu original link here

Solution 3

Recursion Formula:
D(n) = (n-1) [D(n-2) + D(n-1)]

```
public class Solution {
    public int findDerangement(int n) {
        if (n<2) return 0;
        long f[]=new long[n+1];
        f[1]=0;f[2]=1;
        for (int i=3;i<=n;i++) f[i]=(f[i-1]+f[i-2])*(i-1)%1000000007;
        return (int)f[n];
    }
}
```

written by KakaHiguain original link here