Swap Permutation

by wanbo

Problem | Submissions | Leaderboard | Discussions | Editorial

Submissions:474

Max Score:85

Difficulty: Medium

Rate This Challenge:

More

You are given an array *A = [1, 2, 3, ..., n]*:

1. How many sequences *(S_1)* can you get after exact *k* adjacent swaps on *A*?
2. How many sequences *(S_2)* can you get after at most *k* swaps on *A*?

An adjacent swap can be made between two elements of the Array A, A[i] and A[i+1] or A[i] and A[i-1].
A swap otherwise can be between any two elements of the array A[i] and A[j] $\forall$ 1 ≤ i, j ≤ N, i ≠ j.

Input Format

First and only line contains *n* and *k* separated by space.

Constraints

1 ≤ n ≤ 2500
1 ≤ k ≤ 2500

Output Format

Output *S_1 % MOD* and *S_2 % MOD* in one line, where `MOD = 1000000007`.

Sample Input

```
3 2
```

Sample Output

```
3 6
```

Explanation

```
Original array: [1, 2, 3]
1. After 2 adjacent swaps:
We can get [1, 2, 3], [2, 3, 1], [3, 1, 2] ==> S1 == 3

2. After at most 2 swaps:
1) After 0 swap: [1, 2, 3]
2) After 1 swap: [2, 1, 3], [3, 2, 1], [1, 3, 2].
3) After 2 swaps: [1, 2, 3], [2, 3, 1], [3, 1, 2]
==> S2 == 6
```

Current Buffer (saved locally, editable) Java 7

```
1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
```

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

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
    }
}
```

Line: 1 Col: 1

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