Problem 1866: Number of Ways to Rearrange Sticks With K Sticks Visible

Problem Information

Difficulty: Hard
Acceptance Rate: 58.69%
Paid Only: No
Tags: Math, Dynamic Programming, Combinatorics

Problem Description

There are `n` uniquely-sized sticks whose lengths are integers from `1` to `n`. You want to arrange the sticks such that **exactly** `k` sticks are **visible** from the left. A stick is **visible** from the left if there are no **longer** sticks to the **left** of it.

* For example, if the sticks are arranged `[_1_ ,_3_ ,2,_5_ ,4]`, then the sticks with lengths `1`, `3`, and `5` are visible from the left.

Given `n` and `k`, return _the**number** of such arrangements_. Since the answer may be large, return it **modulo** `109 + 7`.

Example 1:

Input: n = 3, k = 2 **Output:** 3 **Explanation:** [_1_ ,_3_ ,2], [_2_ ,_3_ ,1], and [_2_ ,1,_3_] are the only arrangements such that exactly 2 sticks are visible. The visible sticks are underlined.

Example 2:

Input: n = 5, k = 5 **Output:** 1 **Explanation:** [_1_ ,_2_ ,_3_ ,_4_ ,_5_] is the only arrangement such that all 5 sticks are visible. The visible sticks are underlined.

Example 3:

Input: n = 20, k = 11 **Output:** 647427950 **Explanation:** There are 647427950 (mod 109 + 7) ways to rearrange the sticks such that exactly 11 sticks are visible.

Constraints:

* `1 <= n <= 1000` * `1 <= k <= n`

Code Snippets

C++:

```
class Solution {
public:
int rearrangeSticks(int n, int k) {

}
};
```

Java:

```
class Solution {
public int rearrangeSticks(int n, int k) {

}
}
```

Python3:

```
class Solution:
def rearrangeSticks(self, n: int, k: int) -> int:
```