# Problem 629: K Inverse Pairs Array

## Problem Information

**Difficulty:** Hard
**Acceptance Rate:** 48.96%
**Paid Only:** No
**Tags:** Dynamic Programming

## Problem Description

For an integer array `nums`, an **inverse pair** is a pair of integers `[i, j]` where `0 <= i < j < nums.length` and `nums[i] > nums[j]`.

Given two integers n and k, return the number of different arrays consisting of numbers from `1` to `n` such that there are exactly `k` **inverse pairs**. Since the answer can be huge, return it **modulo** `109 + 7`.

**Example 1:**

**Input:** n = 3, k = 0 **Output:** 1 **Explanation:** Only the array [1,2,3] which consists of numbers from 1 to 3 has exactly 0 inverse pairs.

**Example 2:**

**Input:** n = 3, k = 1 **Output:** 2 **Explanation:** The array [1,3,2] and [2,1,3] have exactly 1 inverse pair.

**Constraints:**

* `1 <= n <= 1000` * `0 <= k <= 1000`

## Code Snippets

**C++:**

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

}
};
```

**Java:**

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

}
}
```

**Python3:**

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