Problem 2400: Number of Ways to Reach a Position After Exactly k Steps

Problem Information

Difficulty: Medium
Acceptance Rate: 36.58%
Paid Only: No
Tags: Math, Dynamic Programming, Combinatorics

Problem Description

You are given two **positive** integers `startPos` and `endPos`. Initially, you are standing at position `startPos` on an **infinite** number line. With one step, you can move either one position to the left, or one position to the right.

Given a positive integer `k`, return _the number of**different** ways to reach the position _`endPos` _starting from_`startPos` _, such that you perform**exactly** _`k` _steps_. Since the answer may be very large, return it **modulo** `109 + 7`.

Two ways are considered different if the order of the steps made is not exactly the same.

Note that the number line includes negative integers.

Example 1:

Input: startPos = 1, endPos = 2, k = 3 **Output:** 3 **Explanation:** We can reach position 2 from 1 in exactly 3 steps in three ways: - 1 -> 2 -> 3 -> 2. - 1 -> 2 -> 1 -> 2. - 1 -> 0 -> 1 -> 2. It can be proven that no other way is possible, so we return 3.

Example 2:

Input: startPos = 2, endPos = 5, k = 10 **Output:** 0 **Explanation:** It is impossible to reach position 5 from position 2 in exactly 10 steps.

Constraints:

* `1 <= startPos, endPos, k <= 1000`

Code Snippets

C++:

```
class Solution {
public:
int numberOfWays(int startPos, int endPos, int k) {

}
};
```

Java:

```
class Solution {
public int numberOfWays(int startPos, int endPos, int k) {

}
}
```

Python3:

```
class Solution:
def numberOfWays(self, startPos: int, endPos: int, k: int) -> int:
```