6168. Number of Ways to Reach a Position After Exactly k Steps

My Submissions (/contest/weekly-contest-309/problems/number-of-ways-to-reach-a-position-after-exactly-k-steps/submissions/)

Back to Contest (/contest/weekly-contest-309/)

User Accepted:	5
User Tried:	12
Total Accepted:	5
Total Submissions:	15
Difficulty:	Medium

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** $10^9 + 7$.

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`

JavaScript

```javascript
const ll = BigInt, mod = ll(1e9 + 7), N = 1005;

let fact, ifact, inv;
const comb_init = () => {
    fact = Array(N).fill(0);
    ifact = Array(N).fill(0);
    inv = Array(N).fill(0);
    fact[0] = ifact[0] = inv[1] = 1n;
    for (let i = 2; i < N; i++) inv[i] = (mod - mod / ll(i)) * inv[mod % ll(i)] % mod;
    for (let i = 1; i < N; i++) {
        fact[i] = fact[i - 1] * ll(i) % mod;
        ifact[i] = ifact[i - 1] * inv[i] % mod;
    }
};

const comb = (n, k) => {
    if (n < k || k < 0) return 0;
    return fact[n] * ifact[k] % mod * ifact[n - k] % mod;
};

const numberOfWays = (startPos, endPos, k) => {
    comb_init();
    let res = 0n;
    for (let i = 0; i <= k; i++) {
        let moveRight = i, moveLeft = k - i;
        if (startPos + moveRight - moveLeft == endPos) res += comb(k, i);
    }
    return res;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/791017628/) More Details > (/submissions/detail/791017628/)

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)