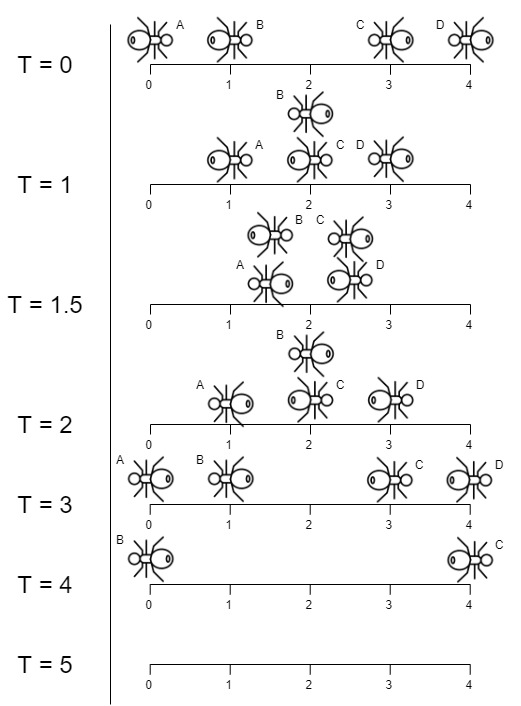
[**Last Moment Before All Ants Fall Out of a Plank**](https://leetcode.com/problems/last-moment-before-all-ants-fall-out-of-a-plank/)

We have a wooden plank of the length n **units**. Some ants are walking on the plank, each ant moves with a speed of **1 unit per second**. Some of the ants move to the **left**, the other move to the **right**.

When two ants moving in two **different** directions meet at some point, they change their directions and continue moving again. Assume changing directions does not take any additional time.

When an ant reaches **one end** of the plank at a time t, it falls out of the plank immediately.

Given an integer n and two integer arrays left and right, the positions of the ants moving to the left and the right, return *the moment when the last ant(s) fall out of the plank*.

Example :

**Input:** n = 4, left = [4,3], right = [0,1]

**Output:** 4

**Explanation:** In the image above:

-The ant at index 0 is named A and going to the right.

-The ant at index 1 is named B and going to the right.

-The ant at index 3 is named C and going to the left.

-The ant at index 4 is named D and going to the left.

The last moment when an ant was on the plank is t = 4 seconds. After that, it falls immediately out of the plank. (i.e., We can say that at t = 4.0000000001, there are no ants on the plank).

Code :

class Solution {

public:

    int getLastMoment(int n, vector<int>& left, vector<int>& right) {

        int mx = 0;

        for(auto i:left){

            mx = max(mx,i);

        }

        for(auto i:right){

            mx = max(mx,n-i);

        }

        return mx;

    }

};