**573. Squirrel Simulation**

* User Accepted: **0**
* User Tried: **0**
* Total Accepted: **0**
* Total Submissions: **0**
* Difficulty: **Medium**

There's a tree, a squirrel, and several nuts. Positions are represented by the cells in a 2D grid. Your goal is to find the **minimal** distance for the squirrel to collect all the nuts and put them under the tree one by one. The squirrel can only take at most **one nut** at one time and can move in four directions - up, down, left and right, to the adjacent cell. The distance is represented by the number of moves.

**Example 1:**

**Input:**

Height : 5

Width : 7

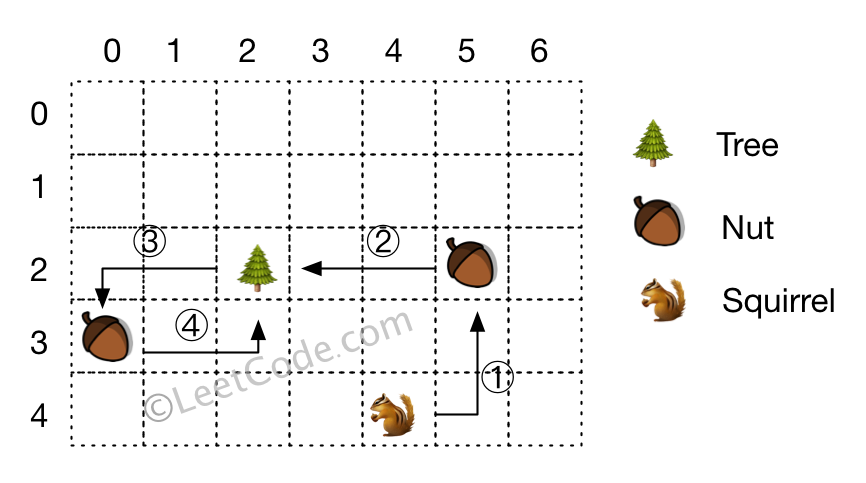
Tree position : [2,2]

Squirrel : [4,4]

Nuts : [[3,0], [2,5]]

**Output:** 12

**Explanation:**



**Note:**

1. All given positions won't overlap.
2. The squirrel can take at most one nut at one time.
3. The given positions of nuts have no order.
4. Height and width are positive integers. 3 <= height \* width <= 10,000.
5. The given positions contain at least one nut, only one tree and one squirrel.

**575. Distribute Candies**

* User Accepted: **504**
* User Tried: **546**
* Total Accepted: **507**
* Total Submissions: **673**
* Difficulty: **Easy**

Given an integer array with **even** length, where different numbers in this array represent different **kinds** of candies. Each number means one candy of the corresponding kind. You need to distribute these candies **equally** in number to brother and sister. Return the maximum number of **kinds** of candies the sister could gain.

**Example 1:**

**Input:** candies = [1,1,2,2,3,3]

**Output:** 3

**Explanation:**

There are three different kinds of candies (1, 2 and 3), and two candies for each kind.

Optimal distribution: The sister has candies [1,2,3] and the brother has candies [1,2,3], too.

The sister has three different kinds of candies.

**Example 2:**

**Input:** candies = [1,1,2,3]

**Output:** 2

**Explanation:** For example, the sister has candies [2,3] and the brother has candies [1,1].

The sister has two different kinds of candies, the brother has only one kind of candies.

**Note:**

1. The length of the given array is in range [2, 10,000], and will be even.
2. The number in given array is in range [-100,000, 100,000].

**572. Subtree of Another Tree**

* User Accepted: **0**
* User Tried: **0**
* Total Accepted: **0**
* Total Submissions: **0**
* Difficulty: **Easy**

Given two non-empty binary trees **s** and **t**, check whether tree **t** has exactly the same structure and node values with a subtree of **s**. A subtree of **s** is a tree consists of a node in **s** and all of this node's descendants. The tree **s** could also be considered as a subtree of itself.

**Example 1:**  
Given tree s:

3

/ \

4 5

/ \

1 2

Given tree t:

4

/ \

1 2

Return **true**, because t has the same structure and node values with a subtree of s.

**Example 2:**  
Given tree s:

3

/ \

4 5

/ \

1 2

/

0

Given tree t:

4

/ \

1 2

Return **false**.

**576. Out of Boundary Paths**

* User Accepted: **149**
* User Tried: **248**
* Total Accepted: **150**
* Total Submissions: **415**
* Difficulty: **Medium**

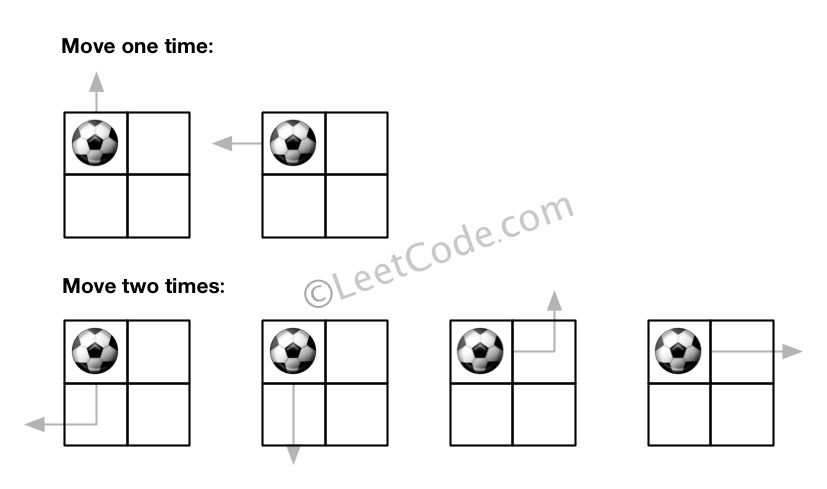
There is an **m** by **n** grid with a ball. Given the start coordinate **(i,j)** of the ball, you can move the ball to **adjacent** cell or cross the grid boundary in four directions (up, down, left, right). However, you can **at most** move **N** times. Find out the number of paths to move the ball out of grid boundary. The answer may be very large, return it after mod 109 + 7.

**Example 1:**

**Input:**m = 2, n = 2, N = 2, i = 0, j = 0

**Output:** 6

**Explanation:**

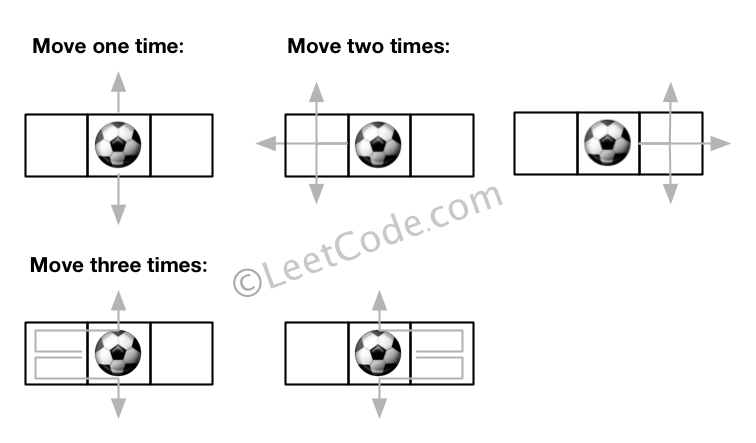


**Example 2:**

**Input:**m = 1, n = 3, N = 3, i = 0, j = 1

**Output:** 12

**Explanation:**



**Note:**

1. Once you move the ball out of boundary, you cannot move it back.
2. The length and height of the grid is in range [1,50].
3. N is in range [0,50].