When the question is like asking to find all the **possible ways / find the best or worst** etc , we can go for recursion.

# Climbing Stairs

**Problem Statement:** Given a number of stairs. Starting from the 0th stair we need to climb to the “Nth” stair. At a time we can climb either one or two steps. We need to return the total number of distinct ways to reach from 0th to Nth stair.

**Brute force** : either we can take steps of 1 stair or 2 stairs , hence the recurrence relation will be very similar to the fibonnacci series.

Time Complexity : O(2^n)

**Best Approach :** Use DP

TC : O(n)

# Frog Jump

**Problem Statement:**

Given a number of stairs and a frog, the frog wants to climb from the 0th stair to the (N-1)th stair. At a time the frog can climb either one or two steps. A height[N] array is also given. Whenever the frog jumps from a stair i to stair j, the energy consumed in the jump is abs(height[i]- height[j]), where abs() means the absolute difference. We need to return the minimum energy that can be used by the frog to jump from stair 0 to stair N-1.

**Approach :**

We have two options: we can jump from the previous stone (or latter one) to the i-th stone, or we can jump from the stone two positions back (or two positions after ith ) to the i-th stone. We choose the option with the minimum cost, where cost is the absolute difference in height between the two stones.

TC : O (2^n)

**Best Approach :** Use DP

TC : O(n)