Stairs

There is an n amount of stairs to climb, in which you can only take one or two steps at a time. What is the total number of different combinations of steps that can be taken to climb the stairs?

Lets list out the possibilities for small inputs:

n = 0 → [0] = 1

n = 1 → [1] = 1

n = 2 → [1,1], [2] = 2

n = 3 → [1,1,1], [1,2], [2,1] = 3

n = 4 → [1,1,1,1], [2,1,1], [1,2,1], [1,1,2], [2,2] = 5

This pattern is similar to Fibonacci numbers, and we can write a recurrence relation for it.

$$T(n) = T(n-1) + T(n-2)$$

Implementation

```
function getTotalCombinations(n){
    if(n <= 0) {
      return 0;
    }
    if(n === 1){
      return n;
    }
    return (this.getTotalCombinations(n-1) +
            this.getTotalCombinations(n-2));
}
```

Complexity

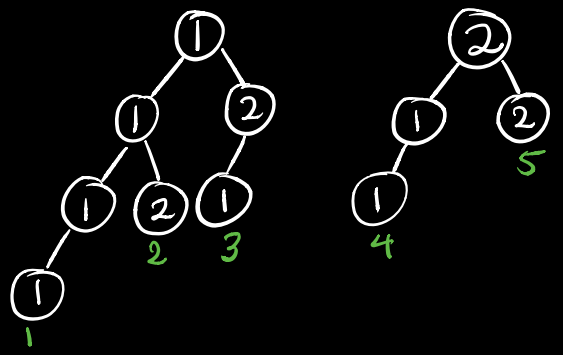
O(n) time

O(n) space due to recursive call stack

Modeling the Problem with Trees

The different stair combinations can also be represented in trees, although this isn't incredibly useful to the solution besides helping find base cases.

$n = 4$



Then count the total number of paths (5)