Problem | Submissions | Comments

Introduction to DP

Medium | Accuracy: 43.72% | Submissions: 3K+ | Points: 4

Geek is learning data structures, and he recently learned about the top-down and bottom-up dynamic programming approaches. Geek is having trouble telling them apart from one another.

When given an integer n, where n is based on a 0-based index, find the n^{th} Fibonacci number.

Every ith number in the series equals the sum of the $(i-1)^{th}$ and $(i-2)^{th}$ numbers, where the first and second numbers are specified as **0** and **1**, respectively.

For the given issue, code both **top-down** and **bottom-up** approaches.

As the answer might be large, return the final answer modulo $10^9 + 7$

Example 1:

C++ (g++ 5.4) | Average Time: 20m | Start Timer

```cpp
// } Driver Code Ends
// User function Template for C++
vector<long long int> dp(100001, 0);
long long int mod = 1e9+7;

class Solution {
  private:
    long long int topDownHelper(int n, vector<long long int> &dpt){
        if(n <= 1) return n;

        if(dpt[n] != -1) return dpt[n];

        return dpt[n] = (topDownHelper(n-1, dpt) + topDownHelper(n-2, dpt)) % mod;
    }

  public:
    long long int topDown(int n) {
        vector<long long int> dpt(n+1, -1);
        return topDownHelper(n, dpt);
    }
    long long int bottomUp(int n) {
        vector<long long int> dpb(n+1, 0);

        dpb[0] = 0;
        dpb[1] = 1;

        for(int i=2;i<=n;i++){
            dpb[i] = (dpb[i-1] + dpb[i-2]) % mod;
        }
        return dpb[n];
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Count ways to reach the n'th stair

Medium Accuracy: 21.9% Submissions: 153K+ Points: 4

There are **n** stairs, a person standing at the bottom wants to reach the top. The person can climb either **1 stair or 2 stairs at a time**. Count the number of ways, the person can reach the top (**order does matter**).

Example 1:

Input:
n = 4
Output: 5
Explanation:
You can reach 4th stair in 5 ways.
Way 1: Climb 2 stairs at a time.
Way 2: Climb 1 stair at a time.
Way 3: Climb 2 stairs, then 1 stair and then 1 stair.

C++ (g++ 5.4) Start Timer

```
// } Driver Code Ends
class Solution
{
    public:
    //Function to count number of ways to reach the nth stair.
    int mod = 1e9+7;
    int countWays(int n)
    {
        int dp[n+1];
        dp[0] = 1;
        dp[1] = 1;
        if(n <= 1){
            return 1;
        }
        for(int i = 2 ; i <= n ; i++){
            dp[i] = (dp[i-1] + dp[i-2])%mod;
        }
        return dp[n];
    }
};


// } Driver Code Ends
```

Custom Input | Compile & Run | Submit