Dynamic Programming

GRASP IITGN

Recursion

1. Base Case (when to stop)
2. Work towards the base Case , assume that the smaller instances of the same problem can be solved (magically)
3. Write the current problem in term of its subproblems

Fibonacci Series

A series of numbers in which every number is sum of the two preceding two numbers.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …

Calculating the nth Fibonacci number

Recurrence Relation

Calculating the nth Fibonacci number

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

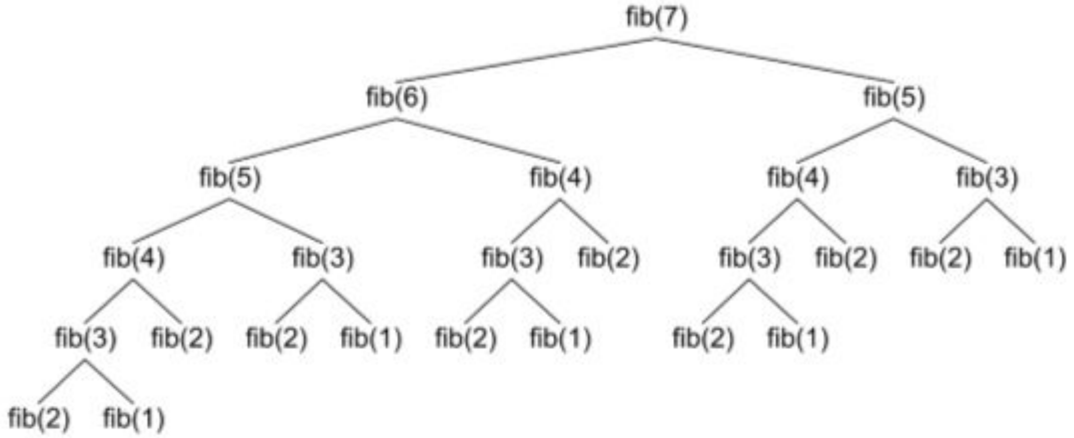
Try the problem

https://practice.geeksforgeeks.org/problems/nth-fibonacci-number/0

Code

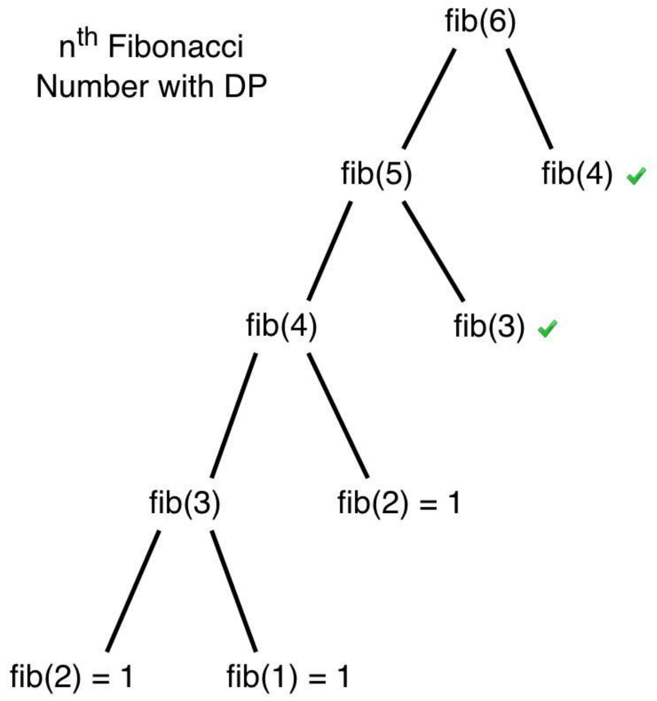
```
def fib(n):
    if n<=2:
        return 1
    return fib(n-1) + fib(n-2)
```

Complexity : O(2^N)



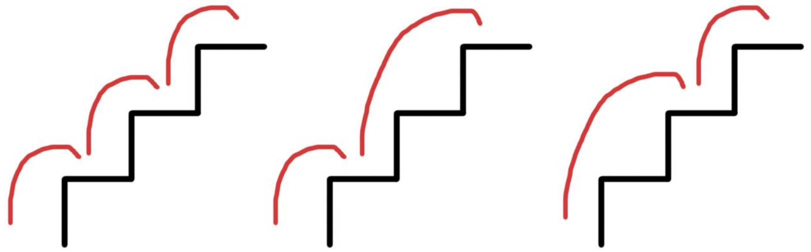
With Dynamic Programming:

O(N)



Stairs Problem

Calculate the number of ways to reach the nth stair. The person can take 1 or 2 steps at a time.



Try the problem

https://leetcode.com/problems/climbing-stairs/

Recurrence Relation

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

Dynamic Programming

Those who cannot remember the past are condemned to repeat it

Use when

1. Optimal Substructure
2. Overlapping Subproblems

Approaches to Dynamic Programming

1. Top-Down Approach (Recursion + Memoization method)

   For solving the given problem, I need to the solutions to its subproblems

2. Bottom-Up Approach (Grid Method)

   First I will solve for subproblems, then will build the solution to larger problem using them
   Fill the grid the same way as the recursive function would have filled it

Coin Change Problem

Find the number of ways to get a change of N rupees in coins, given that you have infinite supply of coins of 1, 2 and 5.

Try the Problem

https://www.hackerrank.com/challenges/coin-change

Recurrence Relation

$F(n)=F(n-1) + F(n-2) + F(n-5)$

How it is different than Stairs Problem

The order of the coins does not matter, but in the stairs problem the order matters.

What matters here is the count of every coin.

Code

0-1 Knapsack Problem

Given the weight limit of a Knapsack as S. There are N items where the value of i^{th} item is Vi and the weight is Wi.

Find the maximum total value that can be put in the knapsack, given the total weight is less or equal to the limit S.

Try the problem

https://www.hackerrank.com/contests/srin-aadc03/challenges/classic-01-knapsack

0-N Knapsack Problem

A variation of 0-1 Knapsack where any item can be chosen any number of times.

Longest Increasing Subsequence

Given a list of N numbers, find the length of the longest increasing subsequence.

A subsequence may be generated by deleting some or no numbers, given the relative order of numbers is same as in the original list.

Try the problem

https://www.hackerrank.com/challenges/longest-increasing-s
ubsequent

An Application of LIS

https://www.geeksforgeeks.org/dynamic-programming-building-bridges/