Dynamic Programming

Dynamic programming is basically an optimization algorithm. It means that we can solve any problem without using dynamic programming but we can solve it in a better way or optimize it using dynamic programming.

The basic idea of dynamic programming is to store the result of a problem after solving it. So when we get the need to use the solution of the problem, then we don't have to solve the problem again and just use the stored solution.

**Dynamic Programming Follows Principle of Optimality**

Where to use Dynamic Programming?

1. Optimal Substructure

A problem is said to have optimal substructure if an optimal solution can be constructed efficiently from optimal solution of its subproblems. Simply stated as expressing a bigger problem into smaller sub-problems.

1. Overlapping Sub-Problems

Same sub-problems occur again and again. So instead of computing for those values again and again we can store them and use it when needed.

Two different ways to store the value: -

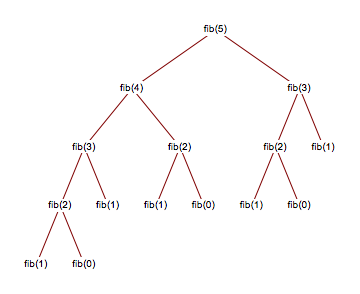
* + - Tabulation (Bottom-Up)
    - Memoization (Top-Down)

Why to use Dynamic Programming?

Let us take an example and understand it.

**Fibonacci Series**

See the recurrence tree give below of Fibonacci series using pure recursion: -



It time complexity is exponential. We reduce the time complexity using dynamic programming. We store the result of a problem after solving it so we don’t have calculate it again and again. We will take a global array and store the result and use it when needed. This reduces the complexity to linear.