L1. Introduction to Dynamic Programming | Memoization | Tabulation | Space optimisation Techniques

08 January 2022
12:09 PM

DP SOLVE BY

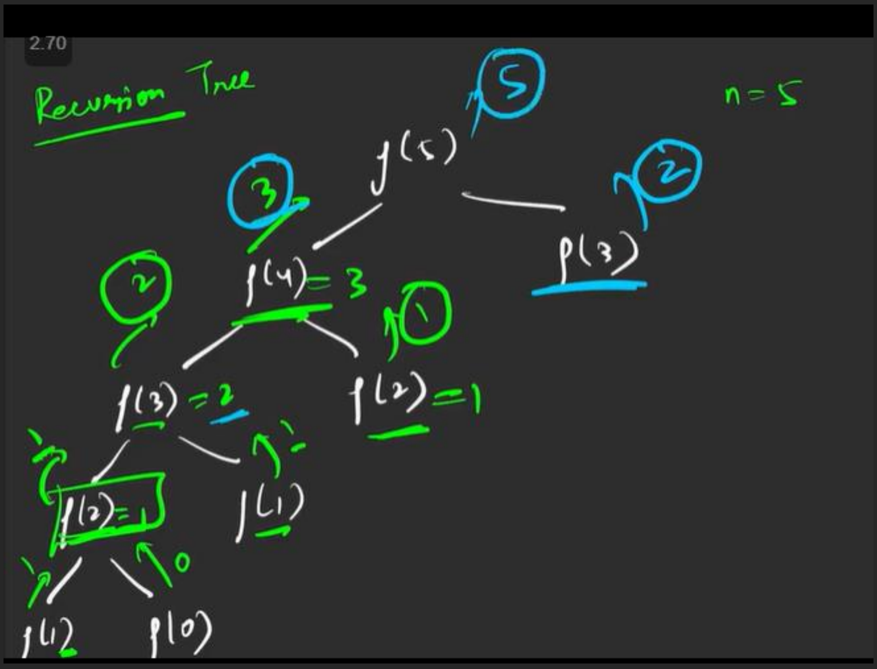
Tabulation --> bottom up
Memoization --> top-down

Memoization --> Tabulation
tc sc tc sc then space optimization

===

Example fibo no.

fibo (5) recursion tree



Overlapping sub problem == when we tends to solve sub problem again use DP i.e do memoization = we store the value of sub problem in some map or table

declare dp[n+1]

step to convert recursion --> dp

1. declare dp array
2. store values for every sub-problem
3. return if curr prob is previously solved

```cpp
int fibo(int n) // Normal recursion
{
    if(n<=1) return n;
    return fibo(n-1) + fibo(n-2);
}


int fibo_dp(int n,vector<int> &dp)
{
    if(n<=1) return n;
    if(dp[n]!=-1) return dp[n]; // (3) return if previously solved
    return dp[n] = fibo_dp(n-1,dp) + fibo_dp(n-2,dp); // (2)store values
}

int main() {
    int n = 5 ;
    vector<int> dp(n+1,-1); // (1) declear dp
    cout<<fibo_dp(n,dp);
}
```

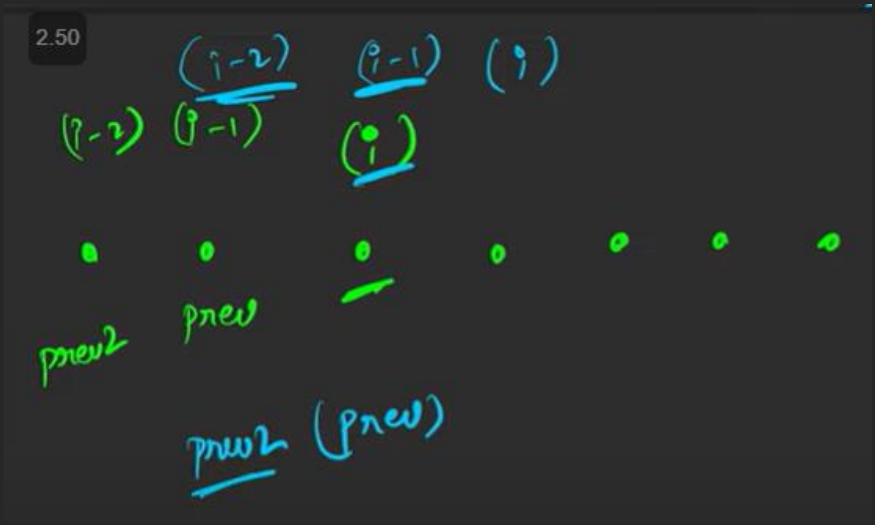
TC => O(n)
SC => O(n) stack space + O(n) dp array

How to convert recursion to tabulation

```cpp
int fibo_tab(int n)
{
    vector<int> dp(n+1,-1);
    dp[0] = 0; dp[1] = 1;
    for(int i=2;i<=n;i++){
        dp[i]=dp[i-1]+dp[i-2];
    }
    return dp[n];
}
```

TC => O(n)
SC => O(n) dp array

More space optimized



TC => O(n)
SC => O(1)

```
int fibo_final(int n)
{
    int prev2 = 0;
    int prev = 1;
    for(int i=2;i<=n;i++)
    {
        int curi=prev+prev2;
        prev2 = prev;
        prev = curi;
    }
    return prev;
}
```

08 January 2022
01:47 PM