

Dynamic Programming

↳ Enhanced Recursion

Pre-requisite → Recursion

Recursion → Memoization → Tabulation

DP

① ↘      ↙ ②

How → DP??

0/1 Knapsack

{

① choices (1/0)

② Optimal solution

↳ (Maxima/Minima)

fibonacci  
series

0	1	2	3	4	5	6	7	8
0	1	1	2	3	5	8	13	21

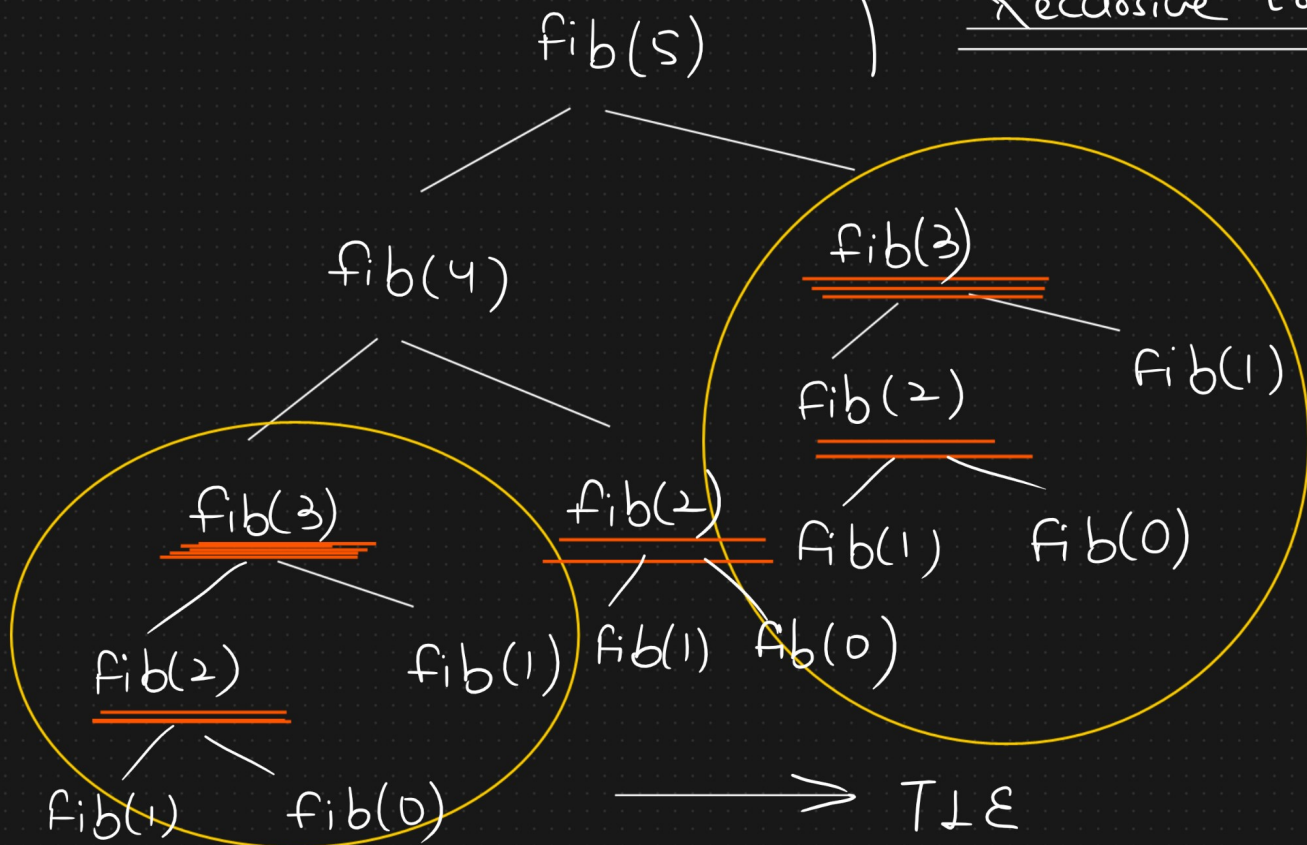
$T(n)$  fib(n) & ① Base case condition  
if ( $n \leq 1$ ) &

c \_\_\_\_\_ return n;

&  
else & ② Recursive function call  
return fib(n-1) + fib(n-2);  
&  
 $T(n-1) + T(n-2)$

&

Overlapping  
Subproblem  
Recursive tree



↳ Time Limit

factorial

Exceeded

fact(n) {

n = 0 or n = 1

return 1

else

return n \* fact(n-1)

}

↓

DP

fact(5)

No need of

DP

|  
5 \* fact(4)

|  
4 \* fact(3)

|  
3 \* fact(2)

|  
2 \* fact(1)

↓  
1

## ① Memoization

↳ Recursion ⇒ Store the  
result of  
every function  
call

## ② Tabulation

↳ No Recursion  
↳ DS to store  
the values  
logically