June 24 Dynamic Programming

24 June 2020   20:59

Fibonacci Series

0th	1st	2	3	4	5	6	7	8

0   1   1   2   3   5   8   13   21

$$fib(n) = fib(n-1) + fib(n-2);$$

Visualizing → Recursion Tree

$f(n)$

$f(n-1)$ $f(n-2)$

$f(n-2)$ $f(n-3)$
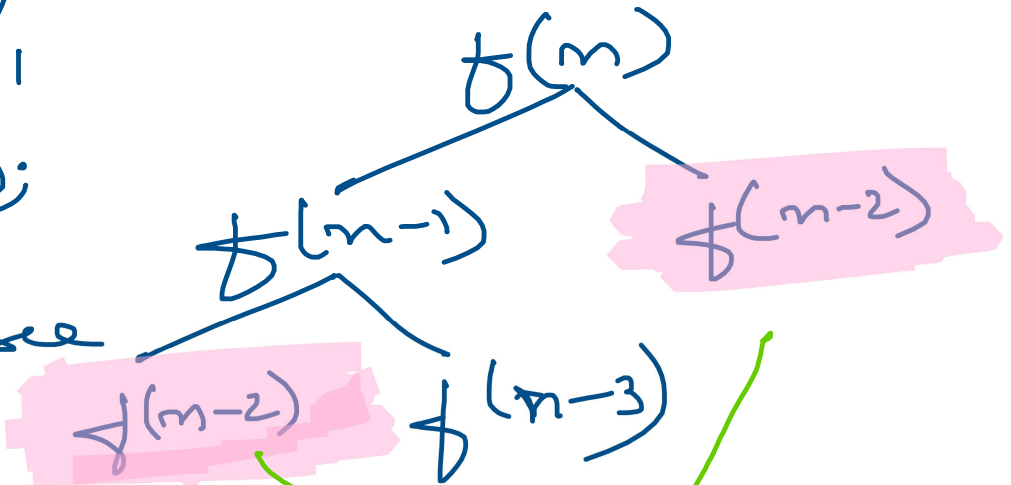
Overlapping Sub Problems

$fib(5)$

$fib(4)$ $fib(3)$

$fib(3)$ $fib(1)$ $fib(2)$ $fib(1)$

$fib(2)$ $fib(2)$ (0) $fib(1)$ $fib(0)$

$fib(1)$ $fib(0)$ $fib(1)$ fib
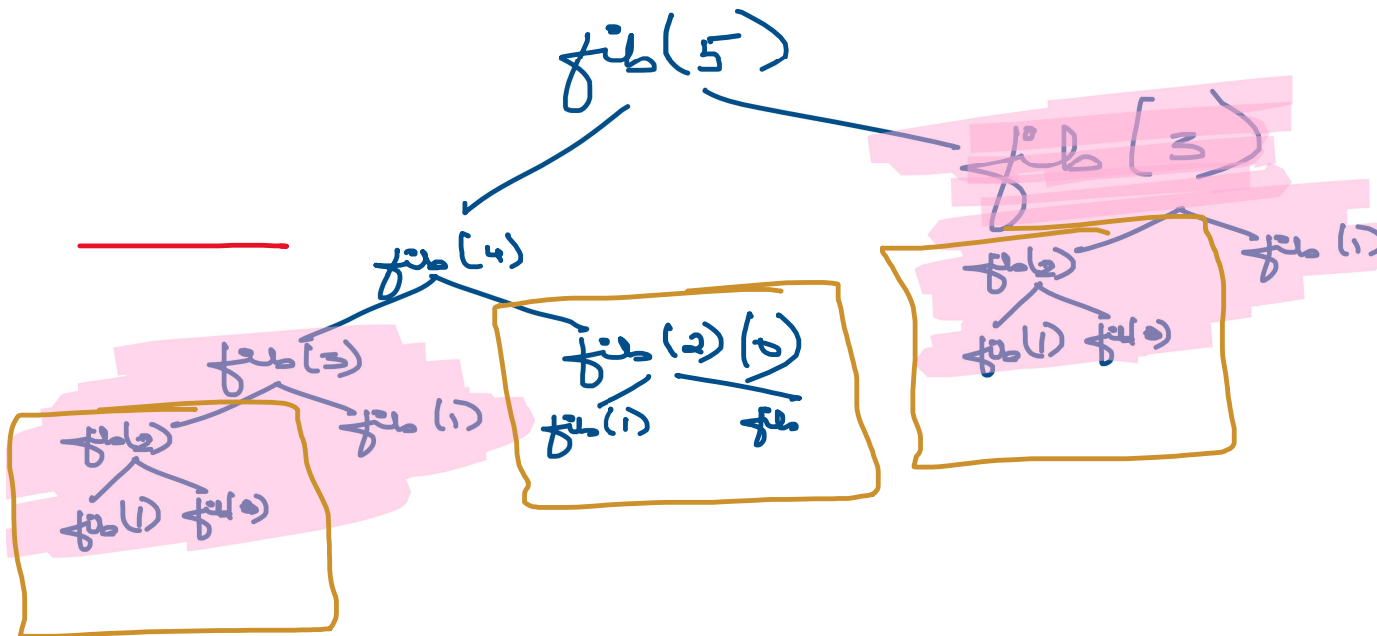
Solution we offer → Store them

& if already stored then don't
recalculate ✓ and pick from storage.

for storage take an array.

0 1 2 3 4 5 6 -- n → index
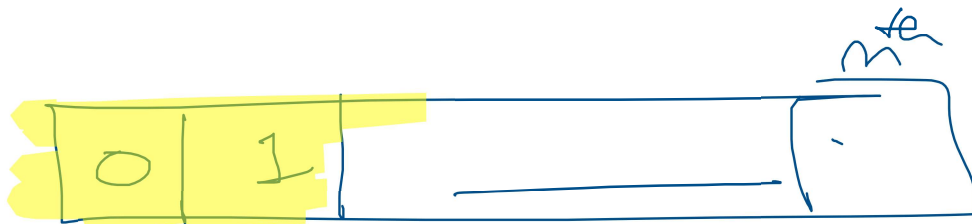
fib(0) ←

Size of Array = ?

= n+1
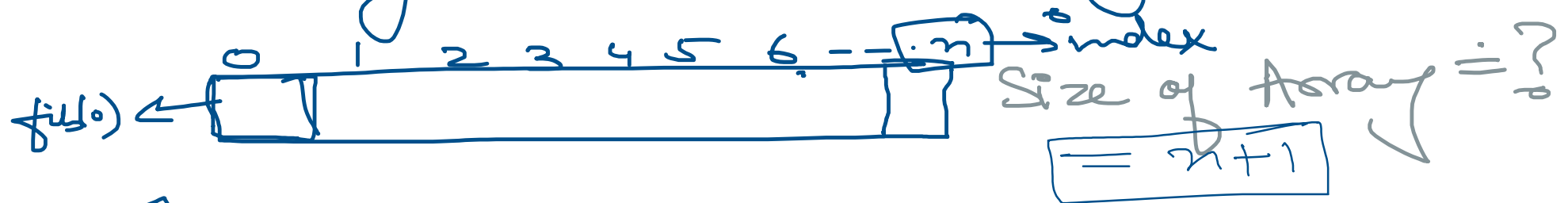
At each array element, fib[index] is stored.

1 → Create array of size [n+1]

2 → Store the answers once calculated

3 → If answer exists then pick from array
do not recalculate

Memoization

Storage → | 0 | 1 | 2 | 3 |

Top-Down Approach

② fib(4)

① fib[3]   fib(2)

fib(1)

1   fib(2)   0

fib(1)   fib(0)

TABULATION

| 0 | 1 | _____ | n-th |
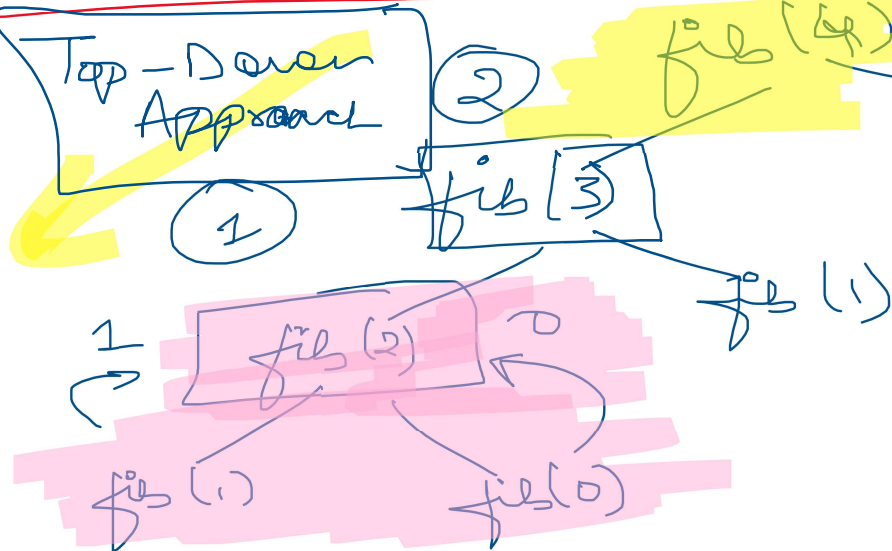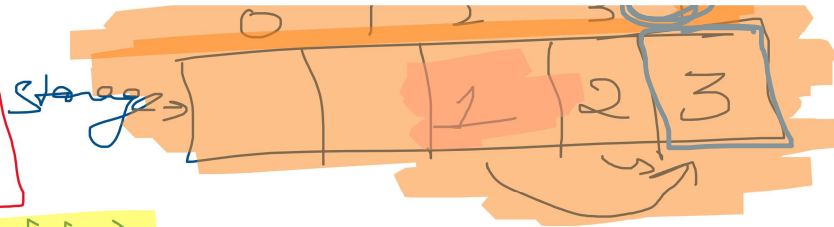
[Bottom → Up] Approach

How to solve any question on DP in 3 steps.
1. Create the Recursive Solution
2. Memoize It
   a. Create a storage array whose dimensions depend on the question & pass it.
   b. Store the results in array.
   c. Check if the function call has already been made and its value is stored in our storage array.
3. Tabulate It
   a. See the memoized solution and tabulate accordingly from bottom to up.