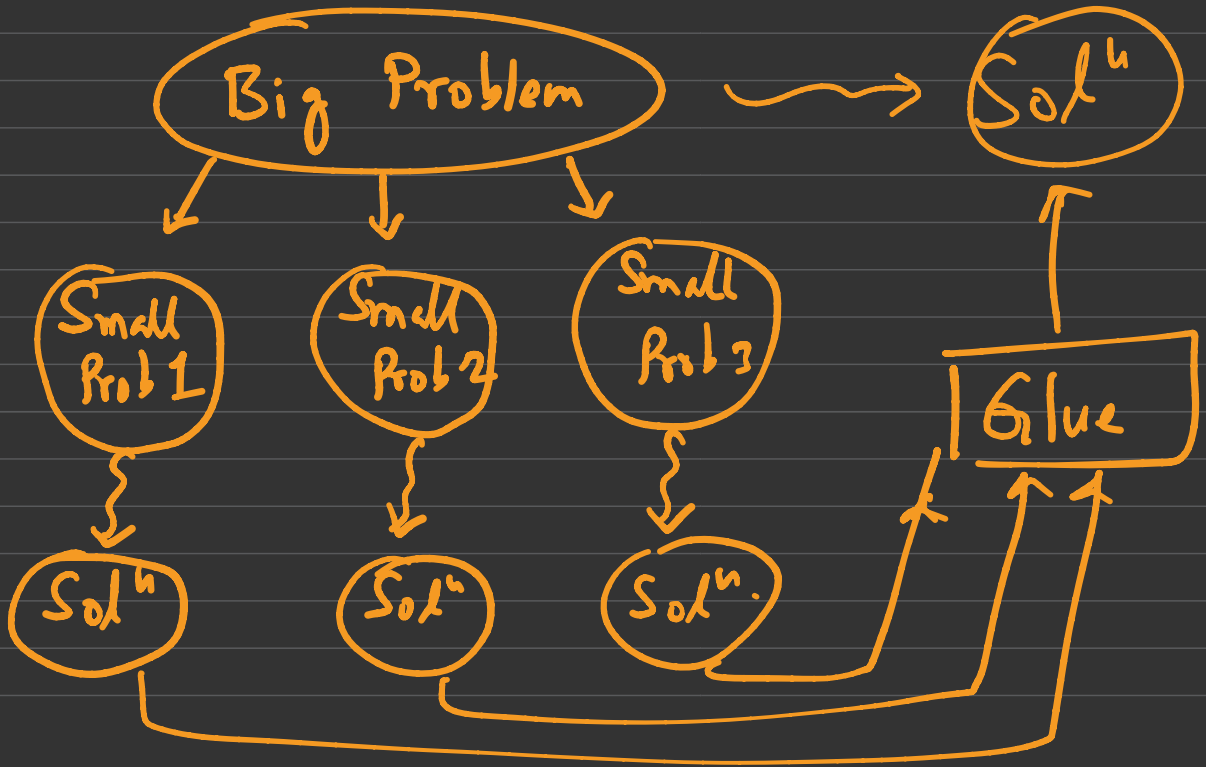
Lecture 01.

Date : 3 Jan 2024

Dynamic Programming



Divide & Conquer

DP

Greedy

Recursion

Induction.

$$Fib(n) = \begin{cases} n & \text{if } n = 0 \text{ or } 1 \\ Fib(n-1) + Fib(n-2) & \text{o/w.} \end{cases}$$

Fib(0) = 0

(1) = 1

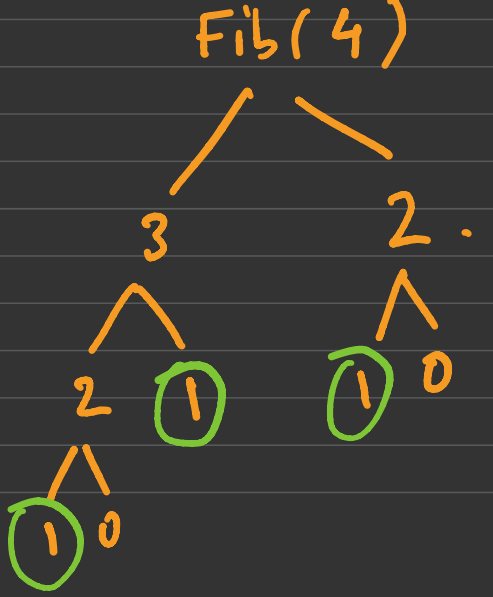
(2) = 1

(3) = 2

(4) = 3

(5) = 5.

```
def Fib(n):
    if n == 0 or n == 1:
        return n
    else: return Fib(n-1) + Fib(n-2)
```



Memoization.

$T(n)$ = # times Fib(1) is called.

$$T(n) = \begin{cases} n & \text{if } n = 0 \text{ or } 1 \\ T(n-1) + T(n-2) & \text{otherwise} \end{cases}$$

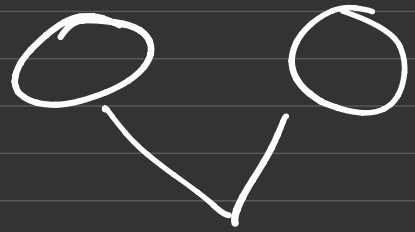
$T(n) = Fib(n)$

$Fib(n) = 2^{O(n)}$ [Fact]

Dynamic Programming Table (DP Table)

A data structure to store the soln to smaller problems.

Overlapping subproblems :-



Subset Sum :-

Input : An array $A[1 \ldots n]$ of $+ve$ integers and a number t.

t ↖ target

Decide:- If there is a subset $S \subseteq [n]$ s.t.

$$\sum_{i \in S} A[i] = t.$$

$[n] := \{1, \ldots, n\}$

$A[1 \ldots n]$: An array with n elements

$A[i \ldots j]$ Subarray with elements $A[i], \ldots, A[j]$

$A = [1, 3, 9, 27]$ $t = 10$

$S = \{1, 3\}$ $t = 11$ ✗

Yes instance

No instance

$A = [1, 2, 4, 8, 16]$

$(A = [1, 3, 9, 27], 10)$ is

$([1, 3, 9, 27], 11)$

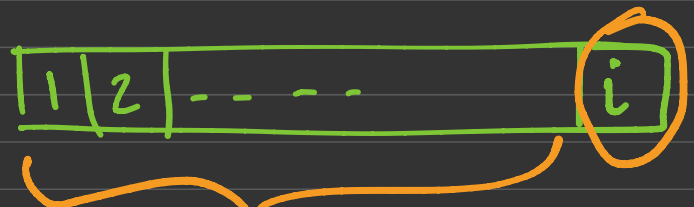
All $1 \leq 31$

give yes instances.

* Remove some element from the array
* Reduce the target.

Subproblems will have the form $(A[1, \ldots, i], j)$

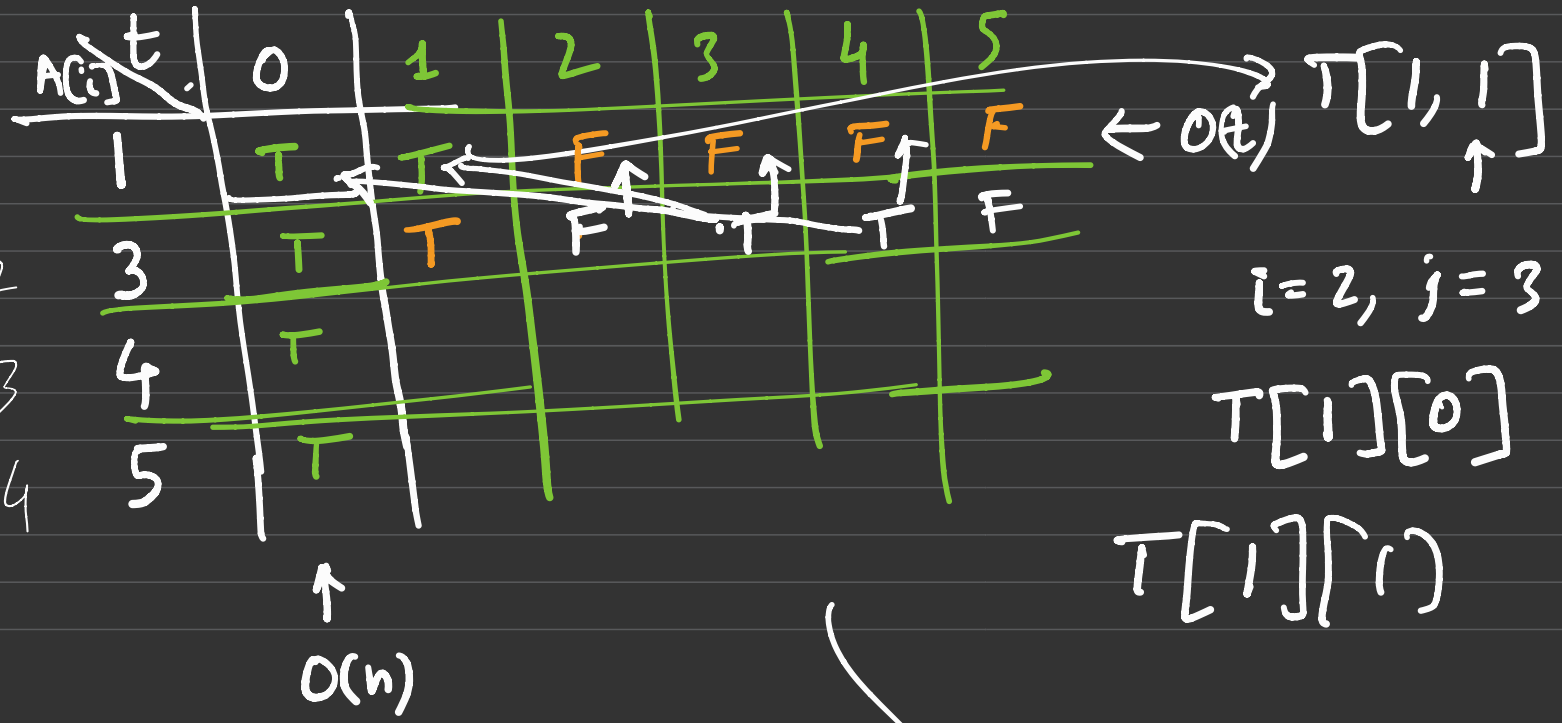
$$T[i][j] = \begin{cases} T & \text{if } (A[1, \ldots, i], j) \text{ is an "yes" instance} \\ F & \text{o/w.} \end{cases}$$



$$T[i][j] = T[i-1][j-A[i]] \vee T[i-1][j]$$

x

$$T[i][j] = \begin{cases} T[i-1][j-A[i]] \vee T[i-1][j] & \text{if } j \geq A[i] \\ T[i-1][j] & \text{o/w} \end{cases}$$



Row = n

col = t

Rest = O(nt)

Total Runtime O(n) + O(t) + O(nt)

O(nt).

Is this polytime ?

— No