unacademy

Miscellaneous - Part II

Complete Course on Algorithms - GATE

Subbarao Lingamgunta • Lesson 53 • May 18, 2023

Dynamic Programming

Applications of DP

1. Fibinocci series
2. Longest common subsequence (LCS)
3. Matrix chain multiplication (mcm)
4. 0/1 knapsack
5. All pairs shortest path
6. Sum of subset problem
7. optimal cost BST
8. TSP

Fibinocci Series

n	0	1	2	3	4	5	6	7	8	9	10
$f(n)$	0	1	1	2	3	5	8	13	21	34	55

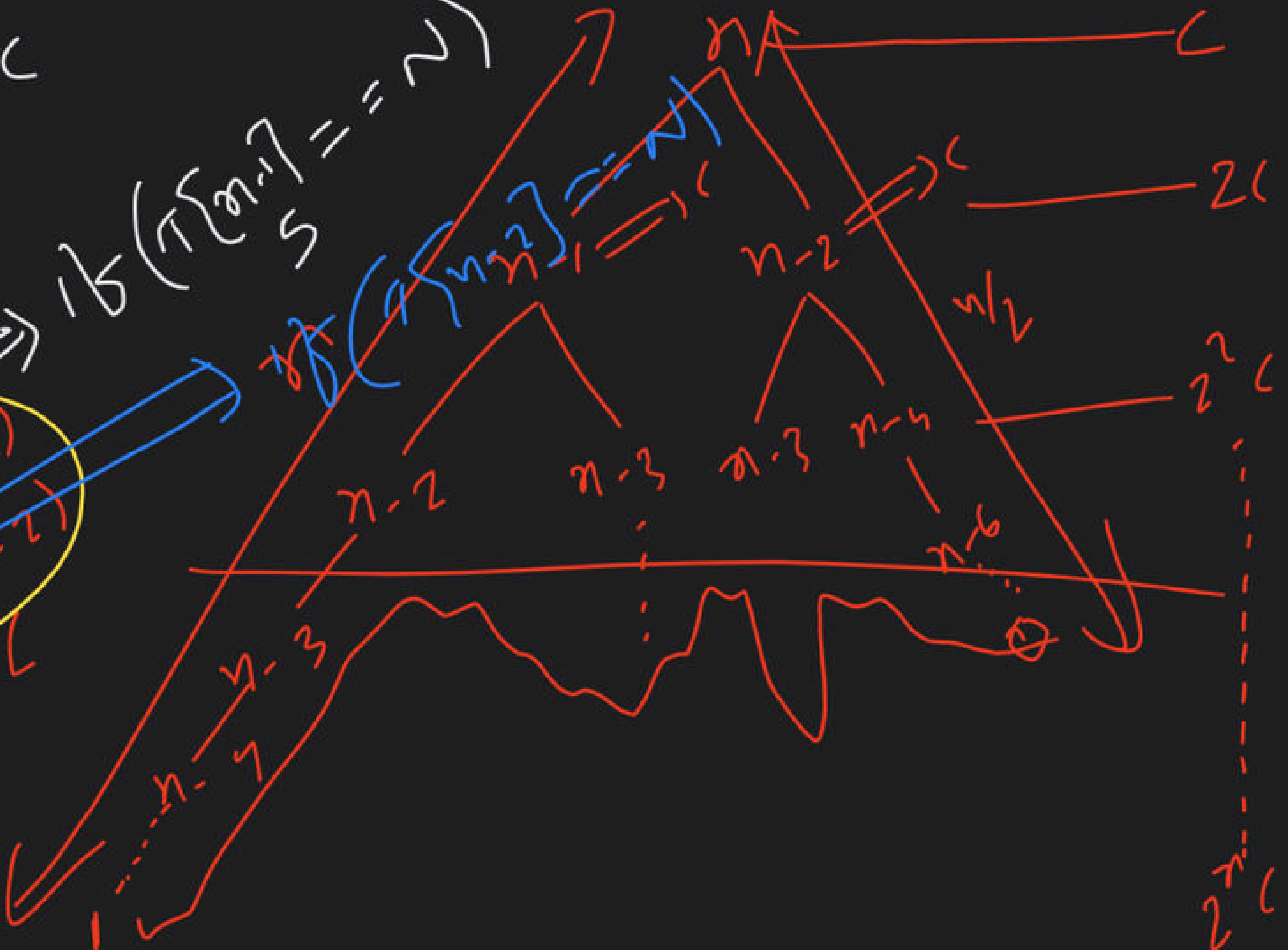
let $T(n) = TC$ of $f(n)$

RR

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

$f(n) \Rightarrow T(n)$

```
f(n)
{
  if (n==0 || n==1)
     return(n)        => c
  else
  {
     a = f(n-1)       => T(n-1)    T[n-1]   if (T[n-1] == N)
     b = f(n-2)       => T(n-2)    T[n-2]   if (T[n-2] == N)
     c = a+b          => c
     return(c)
  }
}
```



$$2^{n/2} < T(n) < 2^n$$

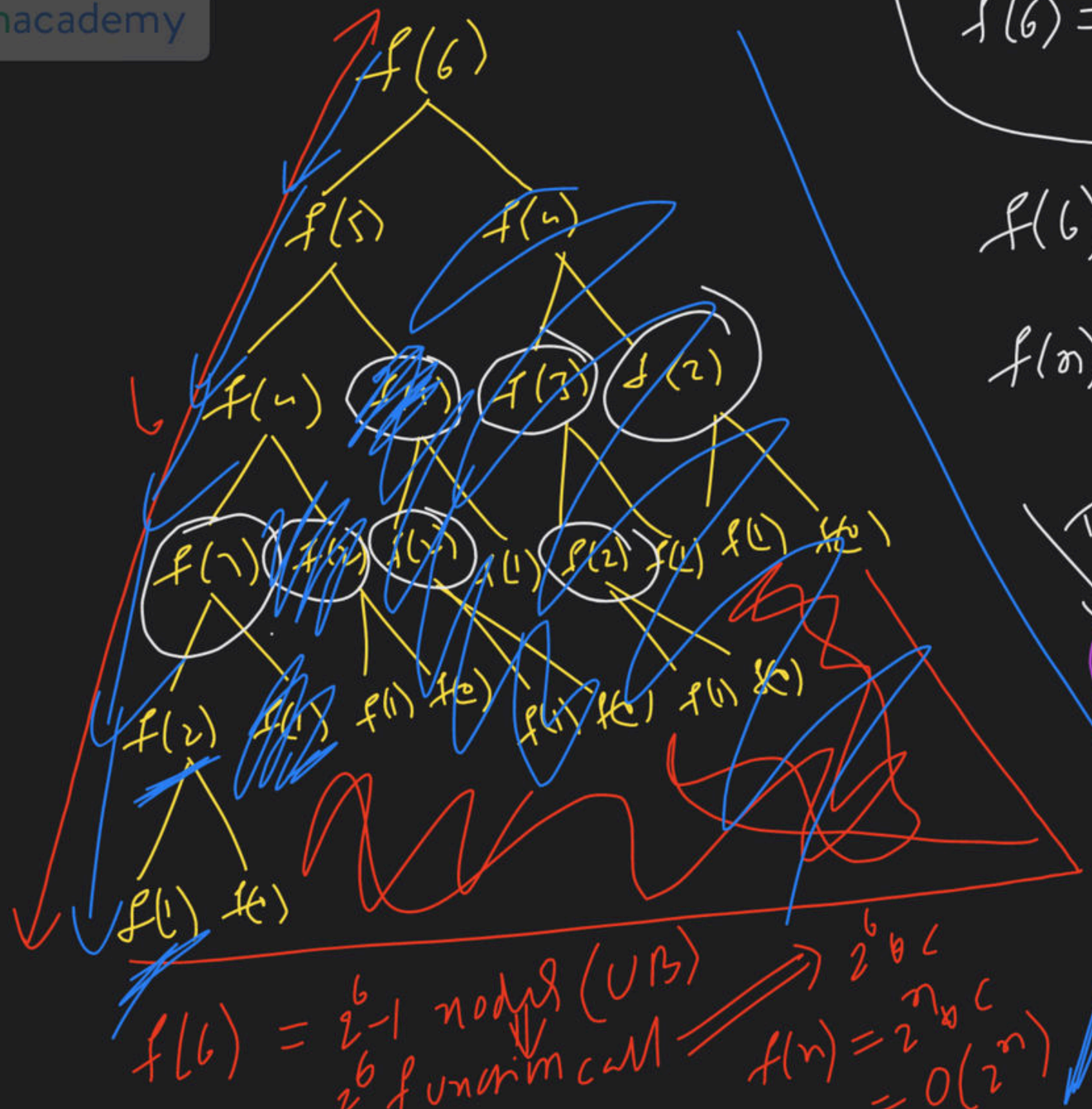
$$\Rightarrow T(n) \le 2^0c + 2^1c + 2^2c \ldots + 2^nc$$

$$\le c\left[\frac{2^n - 1}{2-1}\right]$$

$$\le c \cdot 2^n$$

$$= O(2^n)$$

$$= \Omega(2^{n/2})$$



$f(6) = 2^6 - 1$ nodes (UB)
2^6 function call

$f(n) = 2^n * c$
$= O(2^n)$

$f(6) \Rightarrow f(5), f(4), f(3), f(2), f(1), f(0)$

$f(6) = 6+1$ — distinct function call

$f(n) = n+1$ — function call

$= (n+1) * c \Rightarrow O(n)$ [Time]

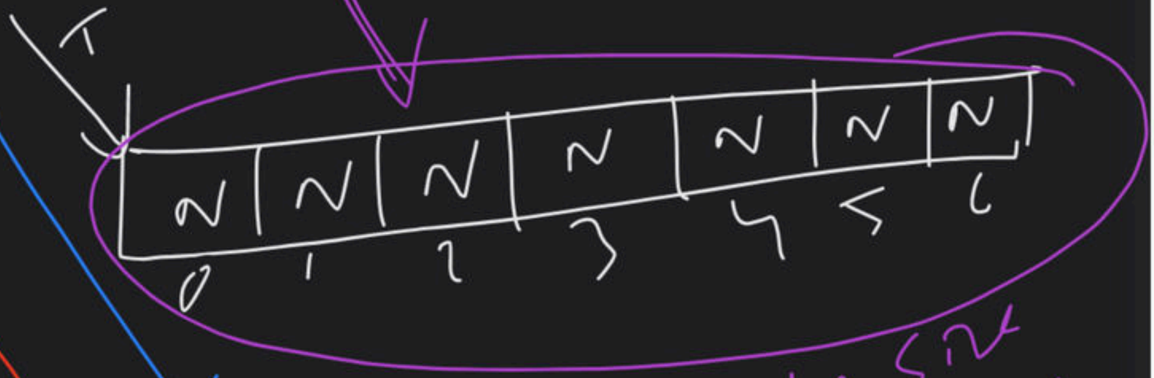


Table size is DFC

```
DP-f(n)
if(n==0 || n==1)
    return n
else
    (1) if(T[n-1]==NI)
    (2) T[n-1] = DP-f(n-1)
    (3) if(T[n-2]==NI)
    (4) T[n-2] = DP-f(n-2)
    (5) T[n] = T[n-1] + T[n-2]
    (6) return(T[n])
```