Wednesday, December 10, 2025 7:33 PM

4 Topics

3 Problem Solving

F(3)
F(5)
F(N) ↓
F(1)

F(5) 120
5 · F(4) 24
4 · F(3) = 6

F(i) → | | 1 | 2 | 6 | 24 | 120 |
0 1 2 3 4 5

F(1) = 1 → Base Case

Memorization + Recursion

F(10)
F(5) → 20

Fibonacci:

0	1	2	3	4	5
0	1	1	2	3	5

f(i) = ?

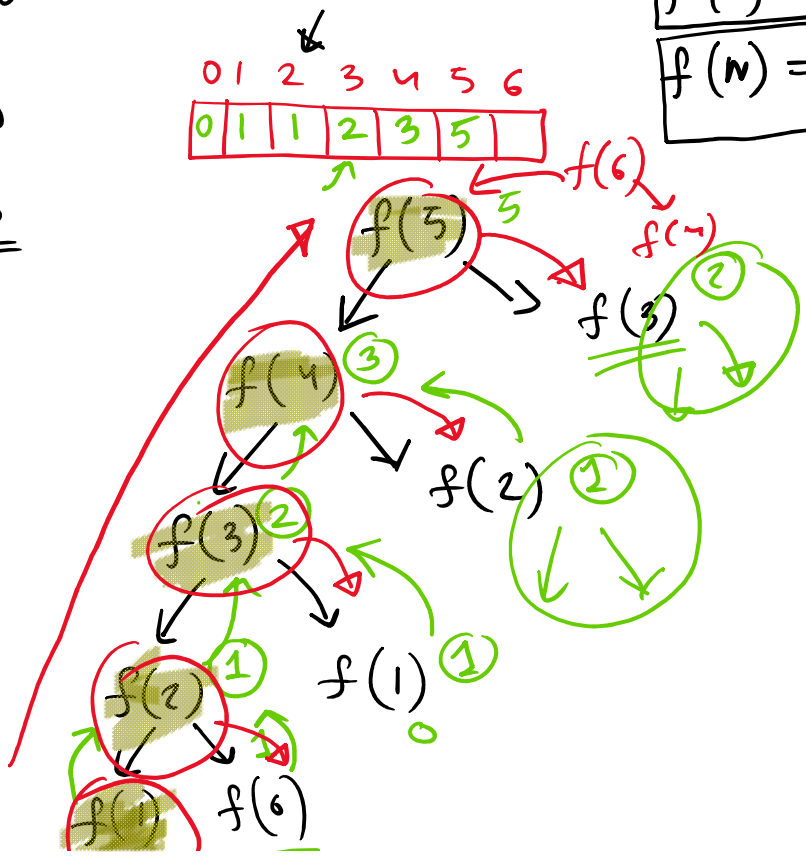
f(0) = 0

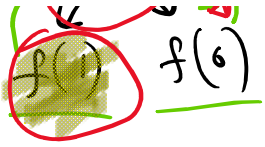
f(4) = 3

Base Case:

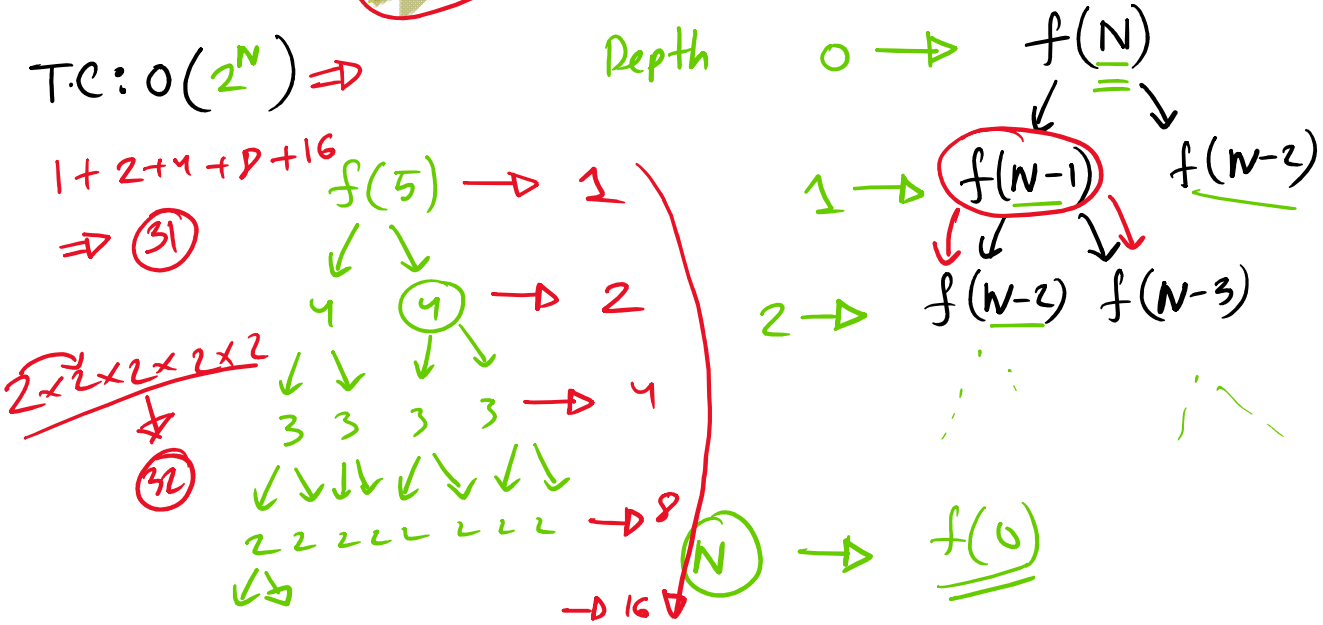
f(1) = 1
f(0) = 0
f(N) = f(N-1) + f(N-2)

f(1000)





T.C : $O(2^N)$ ⇒

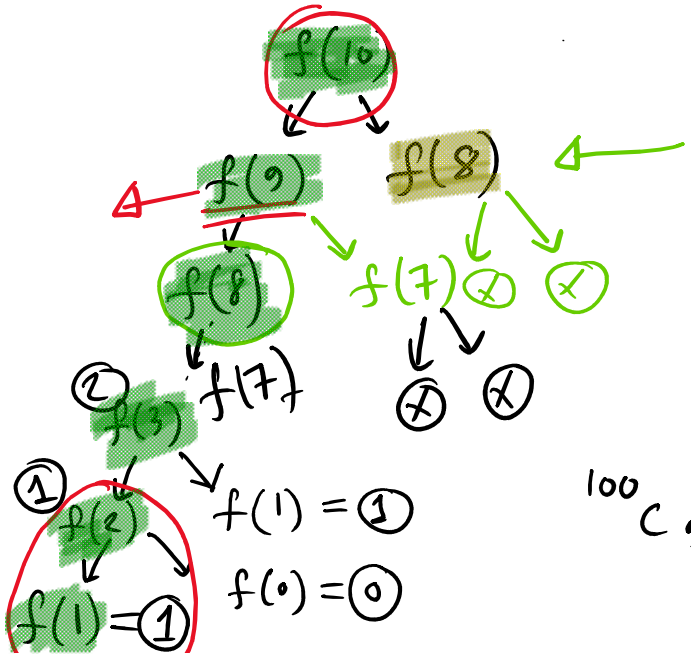


$O(\text{subtree}^N) \approx O(2^N)$

$O(N) \rightarrow$ T.C

$O(N) \rightarrow$ S.C

F(10) =



$O(N) \rightarrow$ T.C

$O(N) \rightarrow$ S.C

$^{100}C_{99} = \frac{100!}{1! \cdot 99!} = 100$

$f(1) = 1$ $\quad f(0) = 0$

$${}^{n}C_{r} = \frac{n!}{(n-r)!\,r!}$$ (1)

(N) টি জিনিস

(R) টি জিনিস বাছাই

$${}^{n}C_{r} = {}^{n-1}C_{r} + {}^{n-1}C_{r-1}$$

${}^{5}C_{2}$ → Counting ✗

${}^{0}C_{0} = 1$

${}^{1}C_{1} = 1$ → Base Case

fib(n)

fact(n)

NCR(n, r)

T.C: O( )

Base Case

ncR(N, 0) = 1

ncR(N, N) = 1

T.C: $O(2^N)$

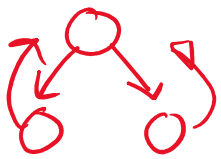
Recursive function

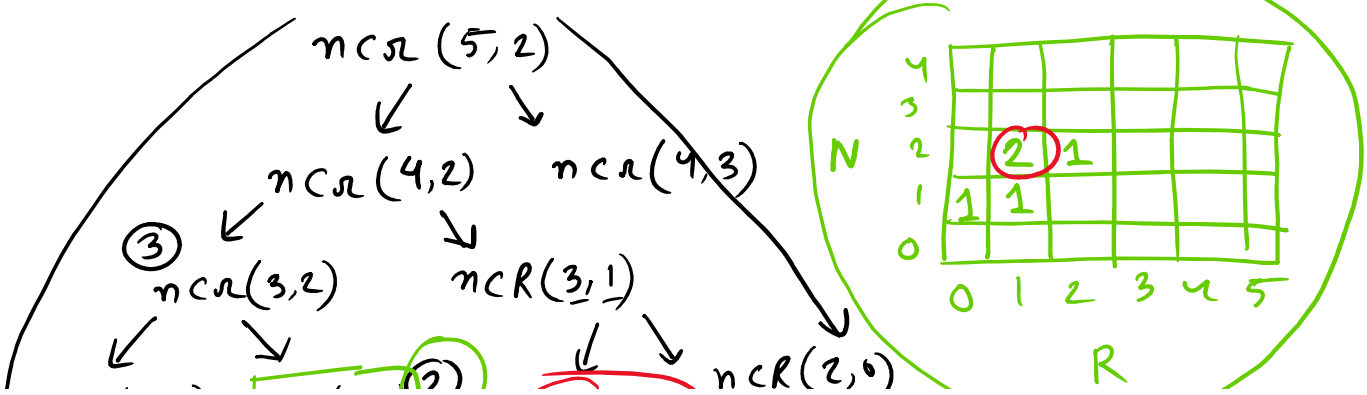
ncR(N, R) = ncR(N-1, R) + ncR(N-1, R-1)

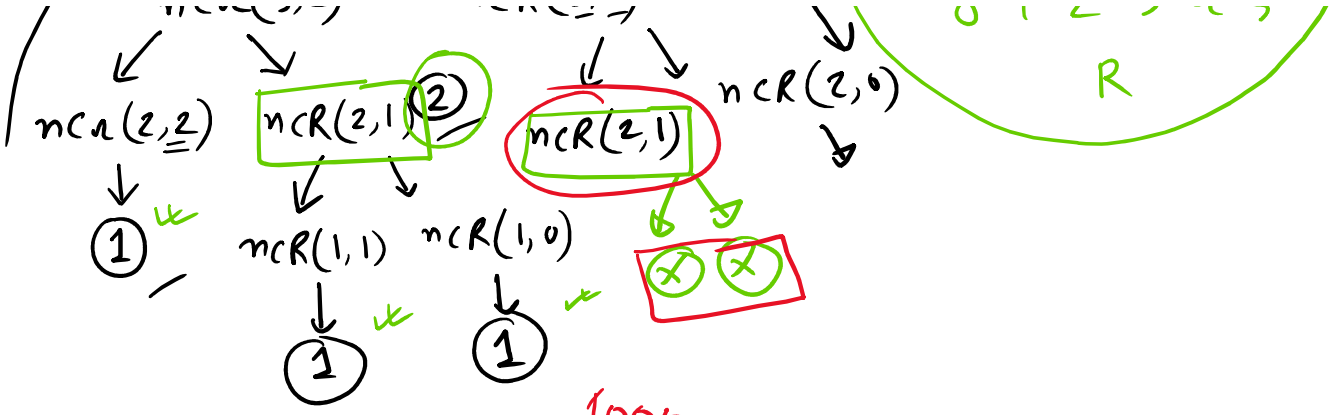
N

N-2

N-3







1006

10

T.C : $O(N \cdot R)$

S.C : $O(N \cdot R + \max(N, R))$

$\approx O(N \cdot R)$