Coding Test - 29th Dec - 31st Dec

3 hour

3 Questions

Recursion → 0, 1, 1, 2, 3, 5, 8, 13

0 1 2 3 4

T_{n+2} T_{n+1} T_n

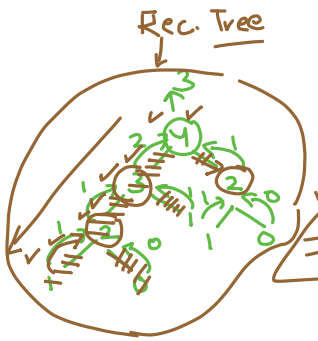
f(0) f(1)

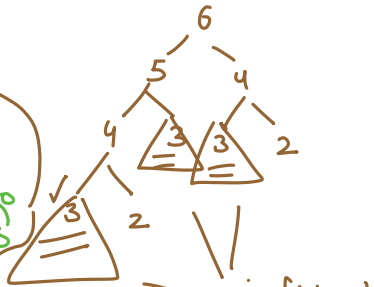
0th Term → 0

1st Term → 1

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

Rec. Tree





4 fn's

4 stack frames.

inefficient

DP

Dynamic Programming

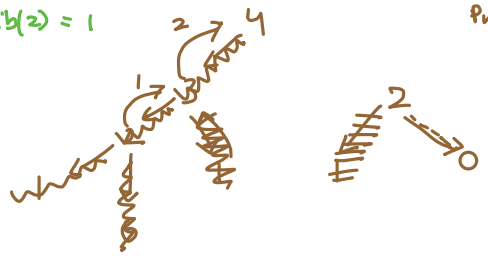
```
def fib(n):
L0  if n==0 or n==1
        return n
L1  f1 = fib(n-1)   ✓ → call
L2  f2 = fib(n-2)   ✓
L3  return f1 + f2
```

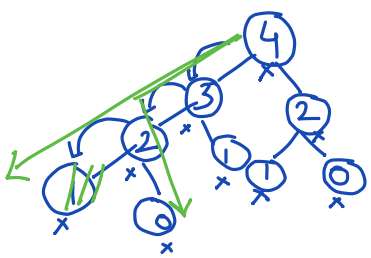

```
{
    return f(n-1) + f(n-2);
}
```

Req1 Req2

Total Calls → Sum of Nodes

Given → Depth.

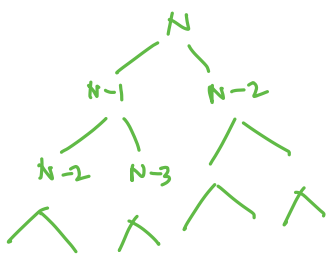




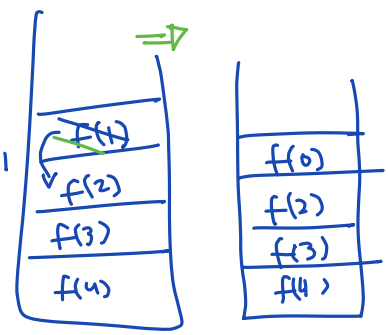
9 calls.

Max calls at an point in stack → 4

time limit



1
2
4
8
⋮
2^{n-1} calls



$= \frac{a(r^n-1)}{r-1} = \frac{2^n-1}{2-1}$

N ⇒ 10 → T

$\sim 2^N$ exponential

N ⇒ 11 → 2T

Tower of Hanoi

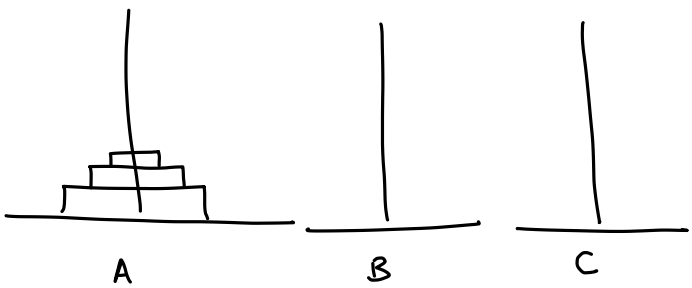
→ 3 Poles (A, B, C)

→ Pole A has disks  N Disks

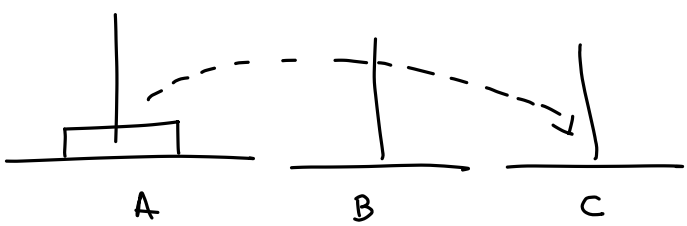
→ Task: Move all disks to other pole (C) using B

constraints:

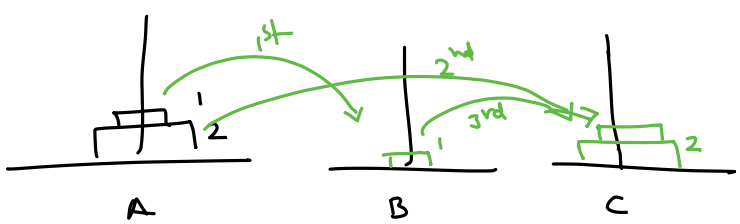
1) 1 Disk at time.
2) Can't put a big disk over smaller disk



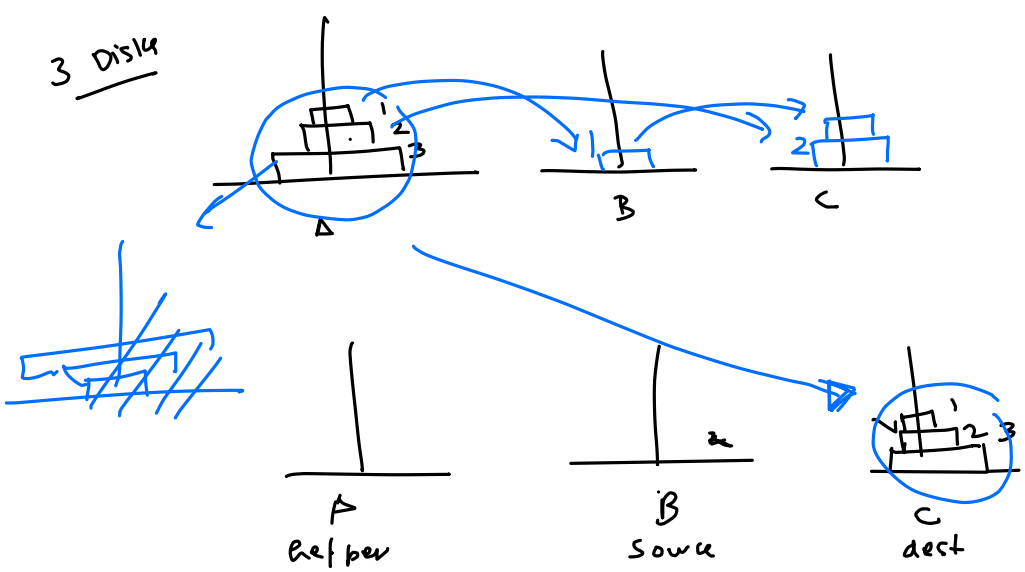
1 Disk



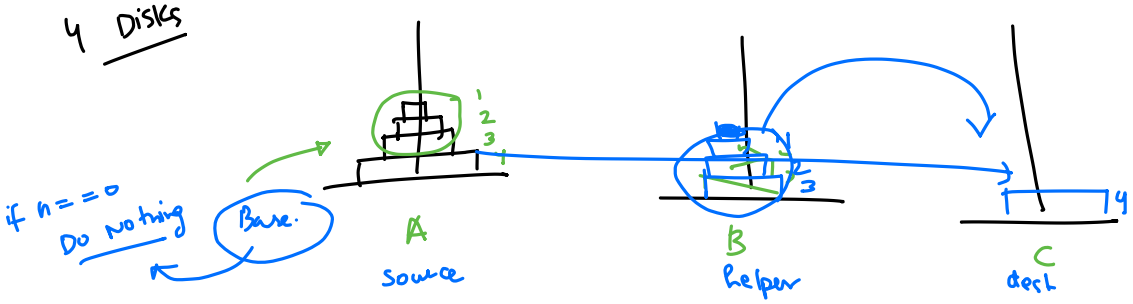
2 Disks



3 Disks



4 Disks

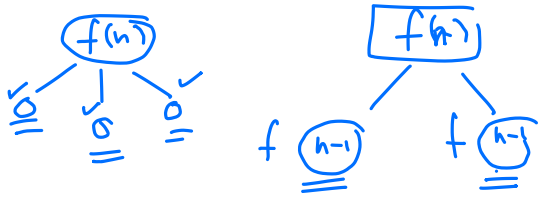


→ Move (N-1 Disks, source to helper, dest)

→ Shift N^{th} disk from some to dest

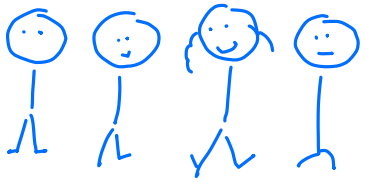
→ Move (N-1 disks, helper, dest, some)

Sub-structure



- N friends →

→ solo
→ couple



A, B, C, D

(A,B), C, D

(A C) (B D)

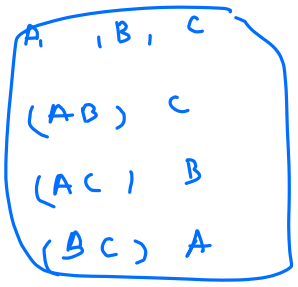
⋮

(A D) B, C

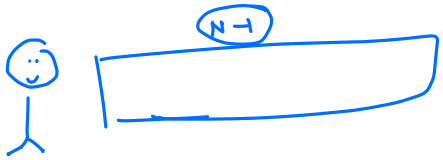
(A D) (B C)

3 friends

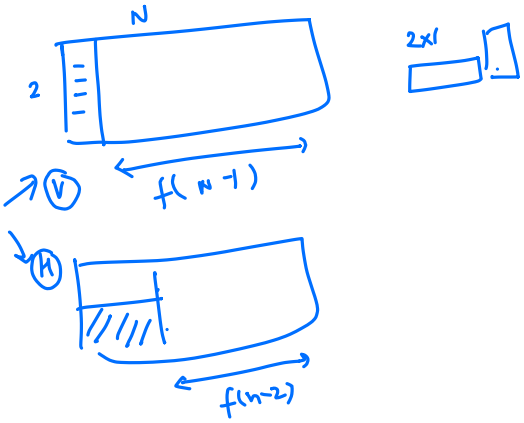
A , B , C

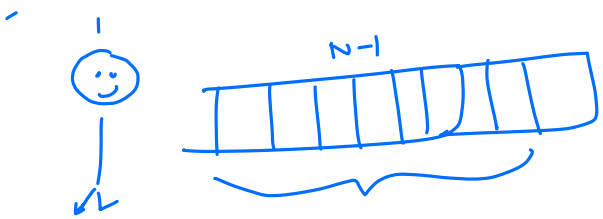


$f(N)$



$f(n)$ = $f(N-1)$ + $f(N-2)$



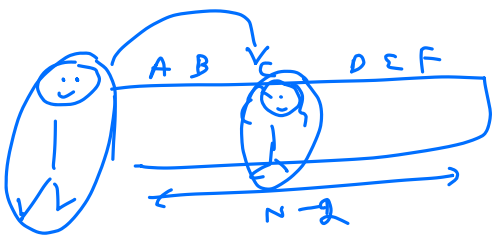


1. and $f(n-1)$

go solo

OR

Solve for 1 Person



$^{N-1}C_1$ and $f(n-2)$

couple

$f(n) = 1 * f(n-1) + (N-1) * f(N-2)$

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

$f(3) = f(2) + 2 \cdot f(1)$

$= 2 + 2 = 4$

$N = 1 \rightarrow 1$

$N = 2 \rightarrow 2$

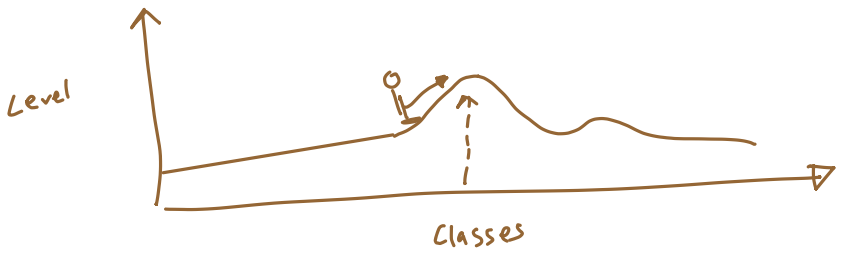
$N = 3 \rightarrow 4$

$f(4) = f(3) + 3 \cdot f(2)$

$= 4 + 3 \cdot 2 =$ **10 ways**

$f(n)$

$f(n) = 1 \cdot f(n-1) + {}^{N-1}C_1 \cdot f(N-2)$



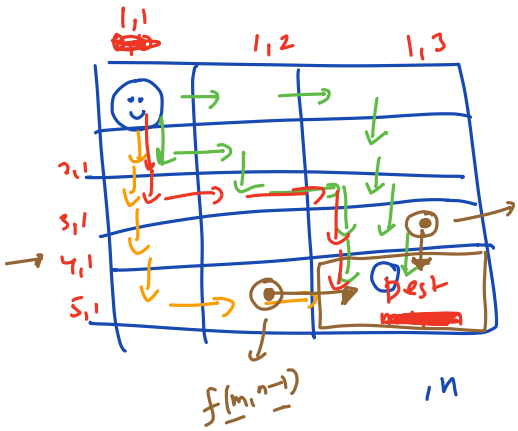
→ Cracking the Coding Interview – Gayel McDowell.

→ Data Structures: Kavumanchi

→ Leet Code / Interview Bit

₹ 1,11,000

Question

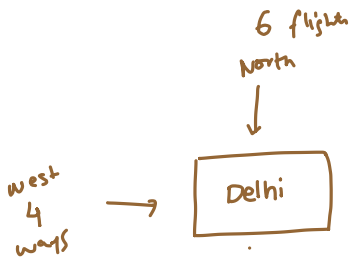


Total paths

move

→ Right ↓ Down

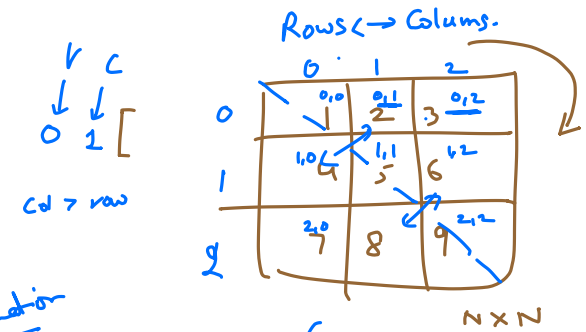
$f(m,n) = f(m-1,n) + f(m,n-1)$

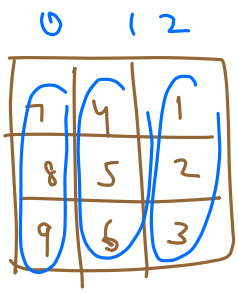


North or West

6 + 4

= 10 ways

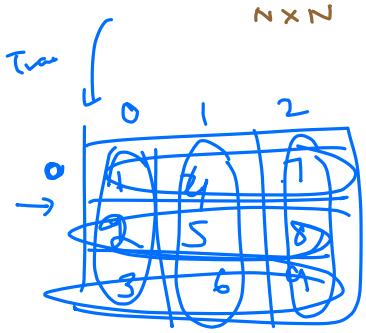




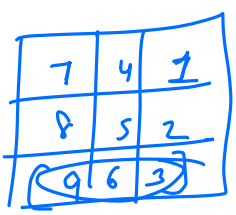
Addition

Observation

Transpose



Rev




```
for ( Rows — )
    for ( colum — ) {
        if col > row :
            a(row)(col) <-> a(col)(row)
    }
}
```