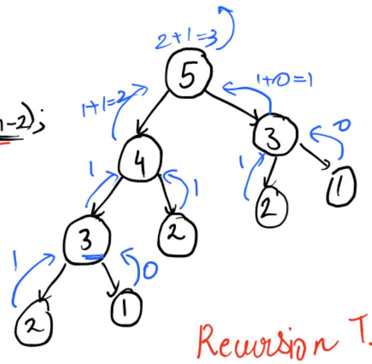
Friday, 9 February 2024
8:20PM

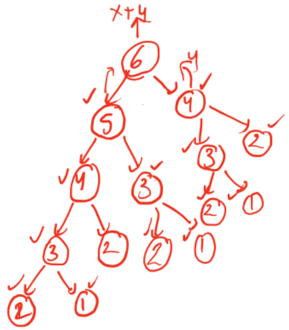
1) Recursive Fibonacci

1 2 3 4 5 6 7 8 9
0, 1, 1, 2, 3, 5, 8, 13, 21, ...

↳ seed

fibonacci(n)
= fibonacci(n-1) + fibonacci(n-2);





BP ⇒ fibonacci(n)

SP ⇒ fibonacci(n-1)
⇒ fibonacci(n-2)

SW ⇒ SP1 + SP2

Base ⇒ seed values

Recursion Tree

→ Visual representation of Recursion

2) Print the pattern

*
* *
* * *
* * * *

n = 4

↳ Print n * in n^{th} row

Recursion

BP = print(n)

SP = print(n-1)

SW = print n *s

BC = if N==0 return

3) Power Calculation

↳ $x, N \Rightarrow X^N = X.X.X.X.X.X \ldots N \text{ times}$

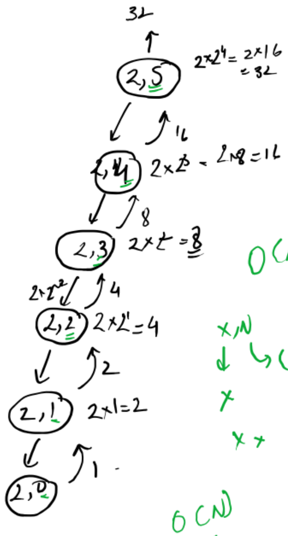
BP

$= X \cdot \underbrace{X^{N-1}}_{SP}$

SW: $X \cdot SP \Rightarrow BP$ ✓

BC: $N==0$

↳ 1



O(N)

x, N ↳ (0 - N)

x

x x O(N)

O(N)

4) Optimised Power Calculation

$X^N = X.X.X.X.X \ldots N \text{ times}$

↳ $X^{N/2} \cdot X^{N/2}$

↳ $X^{N/2+N/2} = X^N$

$a^x \cdot a^y = a^{x+y}$

$2^4 \to 2^2 \cdot 2^2$

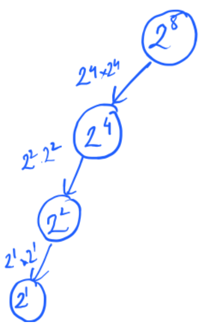
$2^7 \to 2^4 \cdot 2^3$ (SP1, SP2)

⇓

$2 \cdot (2^3 \cdot 2^3)$ $2 \cdot (2^3)^2$ ✓

↳ SP

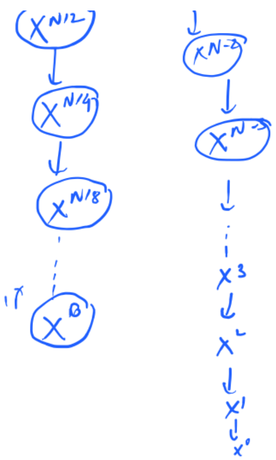
N



$BP = X^N$

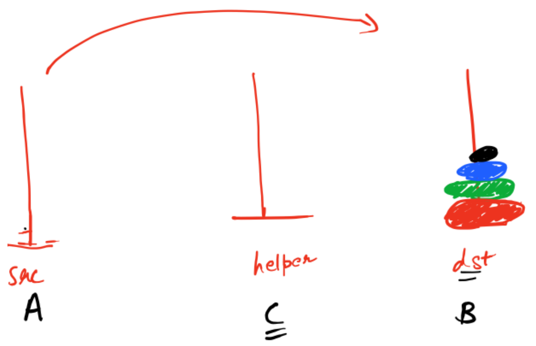
$SP = X^{N/2}$

$N \to \text{even} \to SP \times SP$



SW = ..

N → odd → X . SP . SP



Iterative

for? while? X

BP: 4 from A to B using C

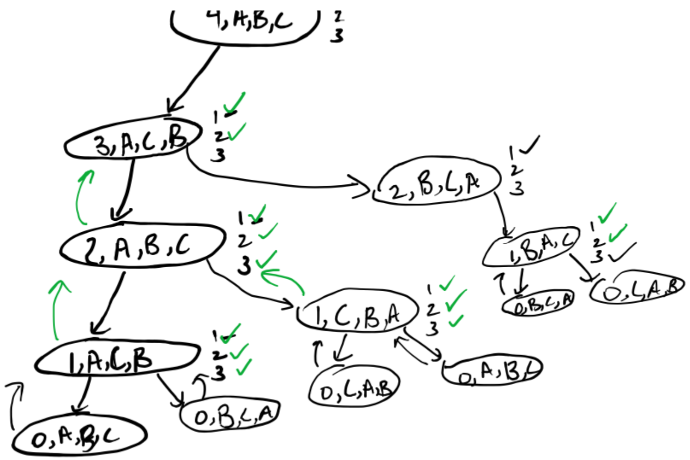
SP: 3 from A to C using B

SW: Move 4th disk from A to B

✓ SP: 3 from C to B using A

BC → move 0 disk return

Recursion ✓



1	A	C
2	A	B
1	C	B
3	A	C
1	B	A