Date:

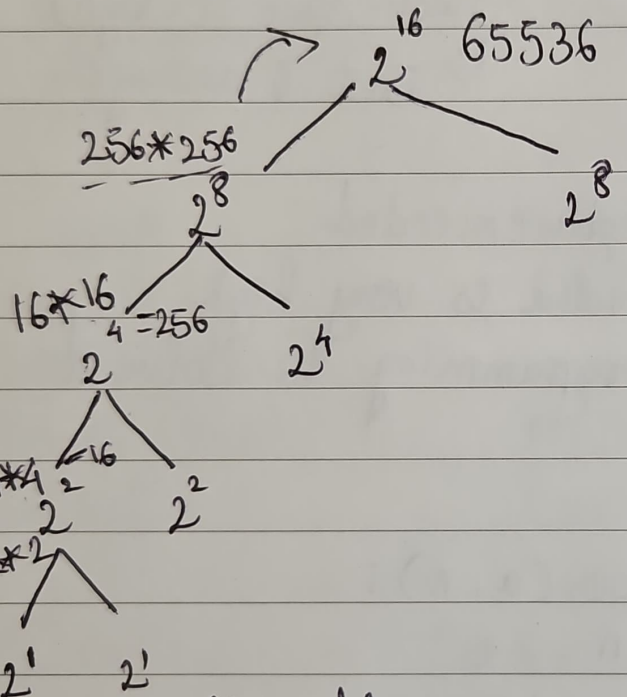
→ Finding of power of An element
↳ Important Interview Question
Amazon → Always

$n=16 \Rightarrow 2^{16}$ → Output

$a=2$ ↳ $2^8 \times 2^8$ → If n is even

Apply Divide & Conquer approach

$2^n = 2^{n/2} * 2^{n/2}$ → n is even



Small problem
↳ n=0

$a^0 = 1$

$a^1 = a$

↳ Small problem

For $2^{17} = 2^{16} * 2$

$= 131072$

Check if n is odd & do one more time multiplication.

$2^{14} = 2^7 * 2^7 = 2^6 * 2$

↳ $2^3 * 2^3$

↳ $2^2 * 2 = 2^1 * 2^1$

Date:

n = -2
a = 2

$2^{-2} \Rightarrow \frac{1}{2^2}$

↳ convert to 1/a & solve.

Skewed Recursive Tree

CBT Recursive Tree

Stack space = O(n)
↳ worst case

Stack space = O(log n)
↳ Best & average

Recursion in Depth exceeded
↳ n value is very high
↳ Dynamic programming to solve it.

Psuedocode:-

findPower(a, n):

Small problem
- n == 0
  ↳ Return 1
- n == 1 → a

n > 1 : mid = n/2

T(n/2) ← b = findPower(a, mid)

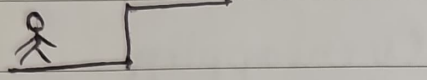
C —
- if n%2 == 0
  return result
- else
  return result * a

Date:

$$T(n) = T(n/2) + c$$
$$= O(\log_2 n)$$

→ Count the number of ways to reach n^{th} stair

2 steps or 1 step at a time :



→ stairs

$n = 1$

Output = 1

$n = 2$

Output = 2

(1, 1) & 2 steps

$n = 4$

Output = 5

(1,1,1,1) (2,1,1)

(1,1,2) (2,2)

(1,2,1)

$n = 3$

(1,1,1) (2,1) (1,2)

In	1	2	3	4	5	6	7
Out	1	2	3	5	8	13	21

↳ Fibonacci series

↳ Sum of last 2 numbers

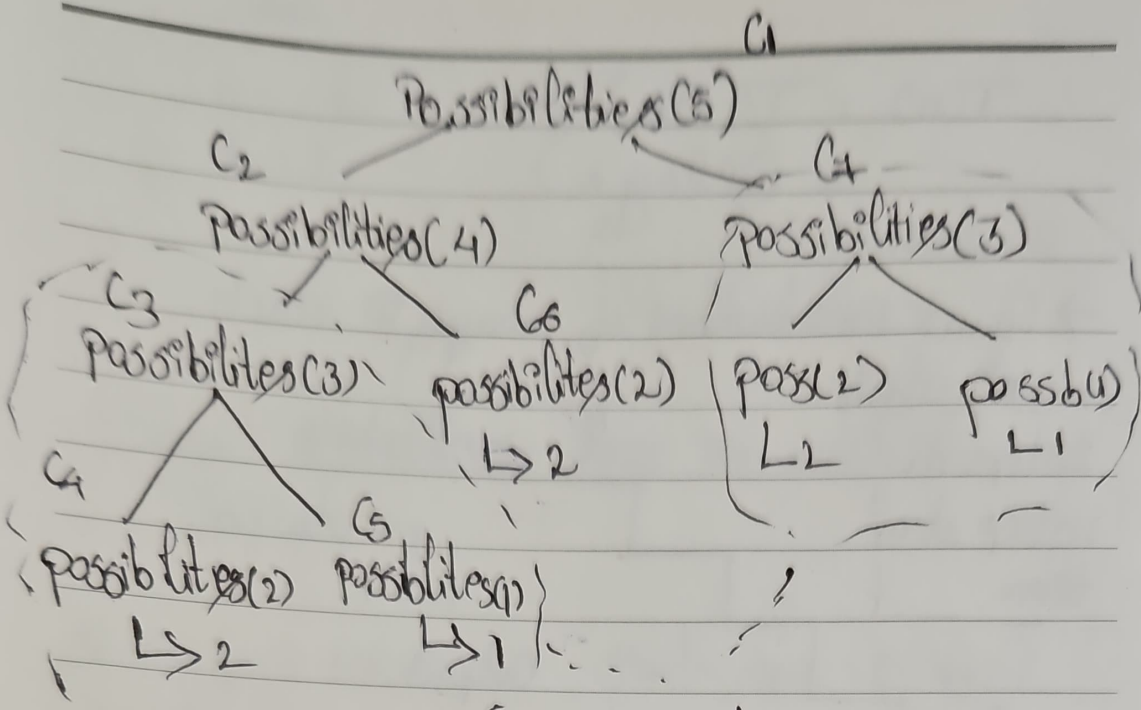
$$way(n) = way(n-1) + way(n-2)$$

$way(n) = fib(n+1)$ } corelation

$ways(1) = fib(2) = 1$

$way(2) = fib(3) = 2$ }

Date:



Overlapping Subproblem

We are solving same method again.
Internal resources are used.
But what if we have n=100000
We will then have many same recursive subtrees are used.

Short notes example.
↳ Traverse & check

Dynamic Programming
↳ Store in an array

Same recursive concept & save it in array or list.