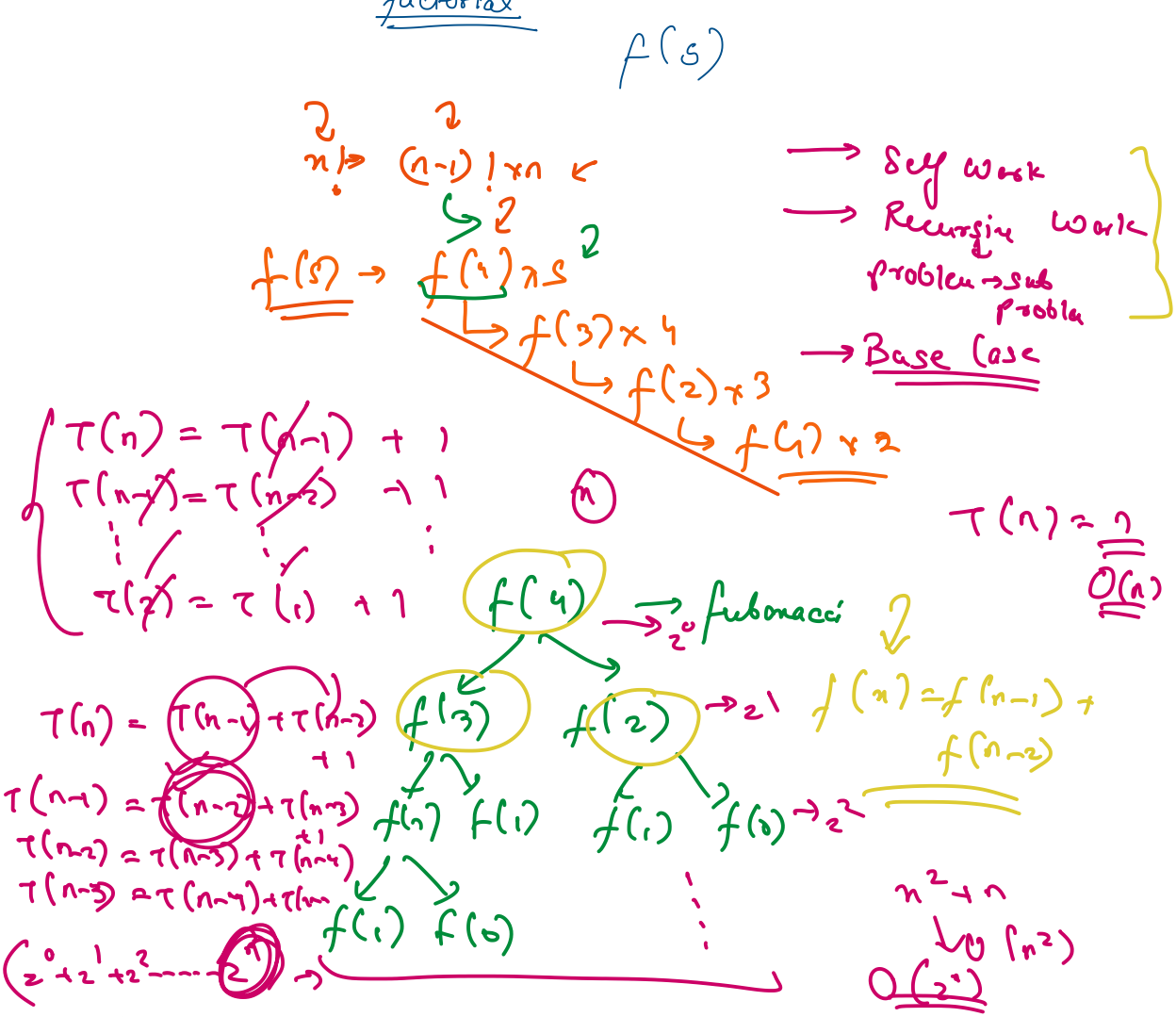
Nagarro class 1

Monday, 3 February 2020 10:48 AM

Recursion

→ When a function calls itself, to do some, repeated tasks, such that it is dividy a problem into sub-problems, along with a memory buffer.

factorial



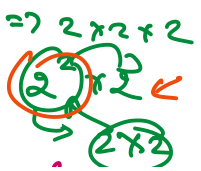
Q ⇒ You will be given 3 numbers, "a", "b" & "c"

→ return → $(a^b) \% c$

Time & Space complexity

a^b

→ $2^3 \Rightarrow 2 \times 2 \times 2$



$a^b = a \times a^{b-1}$

$f(a,b) = a \times f(a, b-1)$

result = 1

for (i = 1 → b) {

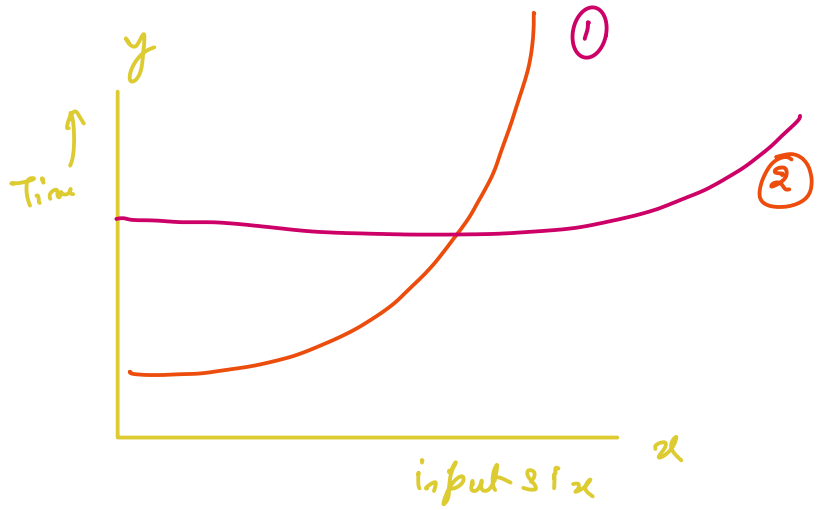
result *= a;

2 → $a \times (a \times f(a, b-2))$ → a^2

a^b

$O(b)$

3

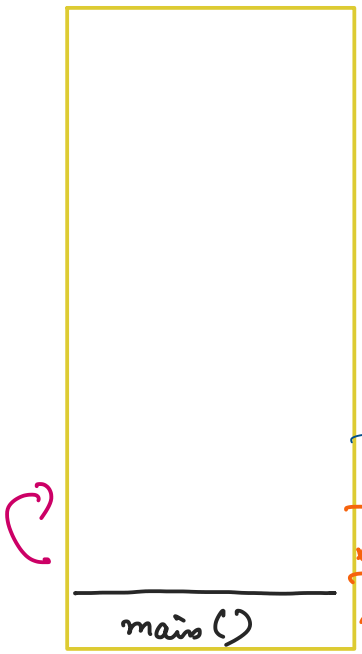


Modulo Arithmetic

$$(a+b) \% c = ((a \% c) + (b \% c)) \% c$$
$$(a-b) \% c = (a \% c - b \% c + c) \% c$$
$$(a \times b) \% c = (a \% c * b \% c) \% c$$

→ Inverse Modulo Arithmetic

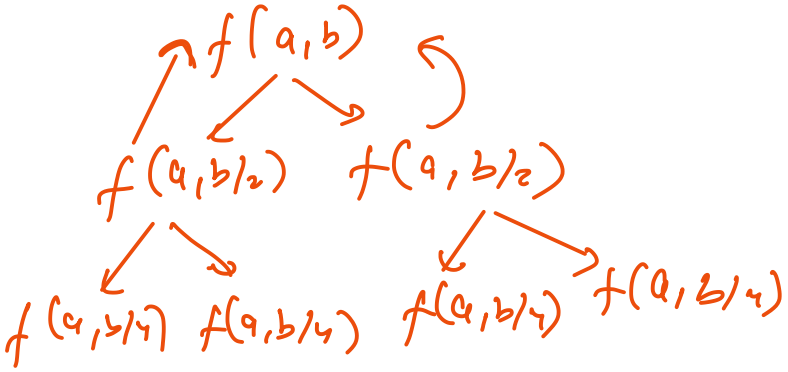
$a^b \% c$




```
1) int pow(int a, int b) {
2)     if(b==1) return a;
3)     int result = pow(a, b-1);
4)     return result * a
5) }
```

2^3

$a^b = a^{b/2} \times a^{b/2}$



$f(a,b) = f(a,b/2) + f(a,b/2) + 1$

$= 2f(a,b/2)$

$\hookrightarrow f(a,b) = 2f(a,b/2)$

$= 2(2 \times f(a,b/4))$

$= 2(2(2 \times f(a,b/8)))$

$= 2 \times 2 \times 2 \cdots\cdots 2 \times f(a,b/2^k)$ (k terms)

Iterations

$\Rightarrow 2^k \times f(a,1) = 2^{\log_2 b} \times 1$

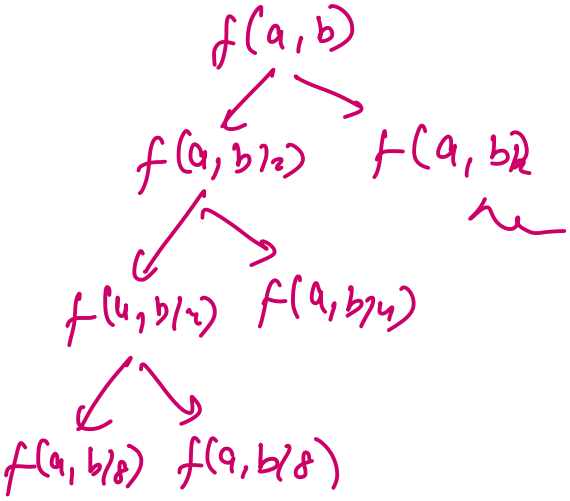
$\Rightarrow b$

$\frac{b}{2^k} = 1$

$b = 2^k$ → taking log both sides

$\log_2 b = (\log_2 2) \times k$

$\rightarrow k = \log_2 b$



$b \rightarrow b/2 \rightarrow b/4 \longrightarrow$

$f(a,b) = f(a,b/2) + 1$

$f(a,b/2) = f(a,b/4) + 1$

$f(a,b/4) = f(a,b/8) + 1$

$f(k) = f(a,1) + 1$

$k \rightarrow$ no. of eq^n

$f(a,b) = f(a,1) + k + 1$

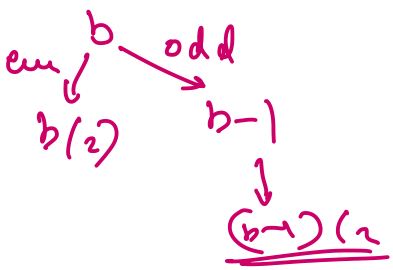
$O(1) \quad \log_2 b$

$T(f(a,b)) = O(\log_2 b)$

$2^{10} = 1024 \approx 10^3$

$e^{20} \approx 10^9$

$2^{30} \approx 10^9$



Q. You have been given a number "n". Count all binary strings of length "n" with no consecutive

ones

n=3

000 101
001 100
010

→ 5

n=1

n=1	n=2	n=3
0	00	000
1	01	001
	10	010
		100
		101
2	3	5

n=4	n=5
8	13

→ n=1 n=2
? ?

2 → 3 → 5 → 8 → 13 - - - -

Q→ friends pairing → n

n=3 A B C

(A) (B) (C)
(AB) (C)
(BC) (A)
(AC) (B)

A B C D → f(4)
→ f(3)

(B) (C) (D)
(BC) (D)
(CD) (B)
(BD) (C)

f(n-2)
?
(AB) (C)(D)
(AB) (CD)

(A) (B)(C)(D)
(A)(BC) (D)
(A)(CD) (B)

(A) (B) (C)

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

f(3)

BCD

f(2)

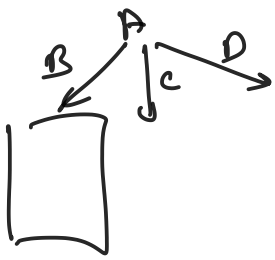
(B) (C) (D)

(B) (CD)

$f(3) = f(2) + \_\_\_\_$

f(2)

(n-1)



(AB) [C, D]

(AB) (C) (D)

(AB) (CD)

(AC) (B) (D)

(AC) (BD)

(AD) (B) (C)

(AD) (BC)

AB

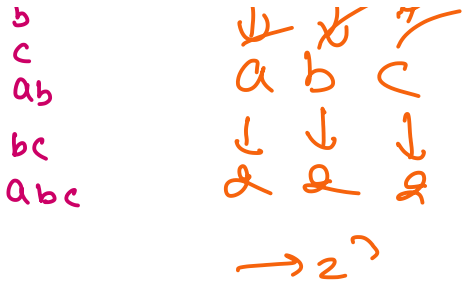
(C) (D) (E)

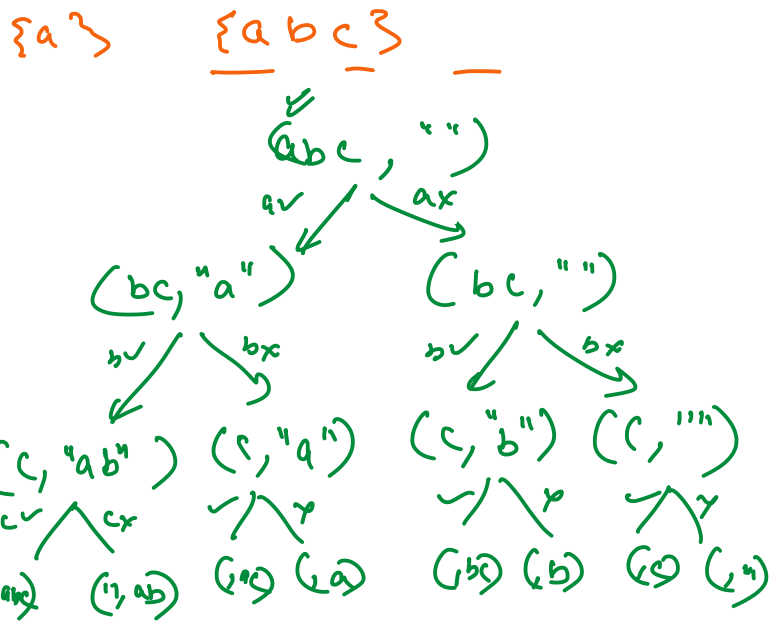
(CD) (E)

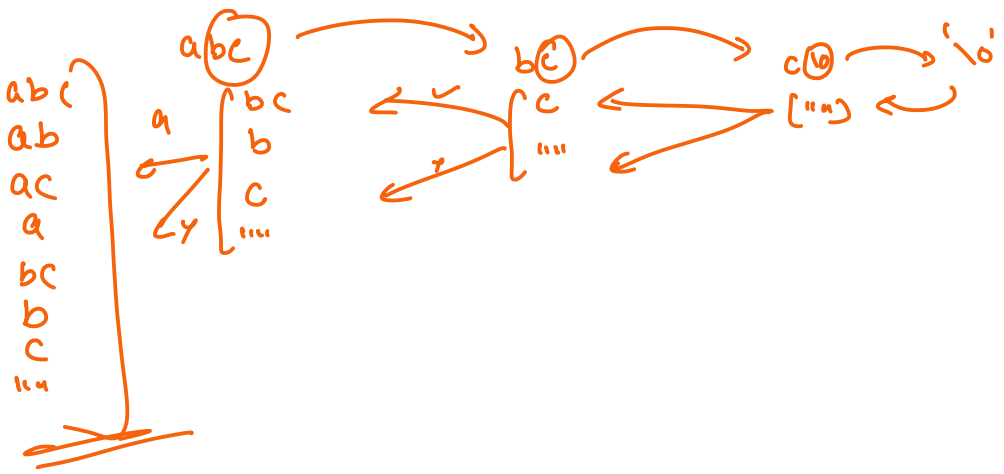
(C) (DE)

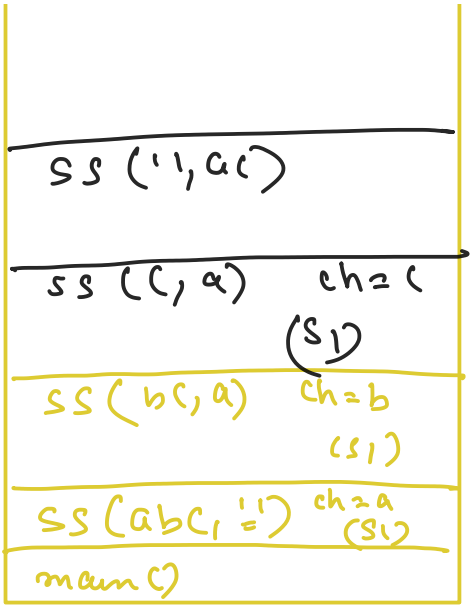
(CE) (D)

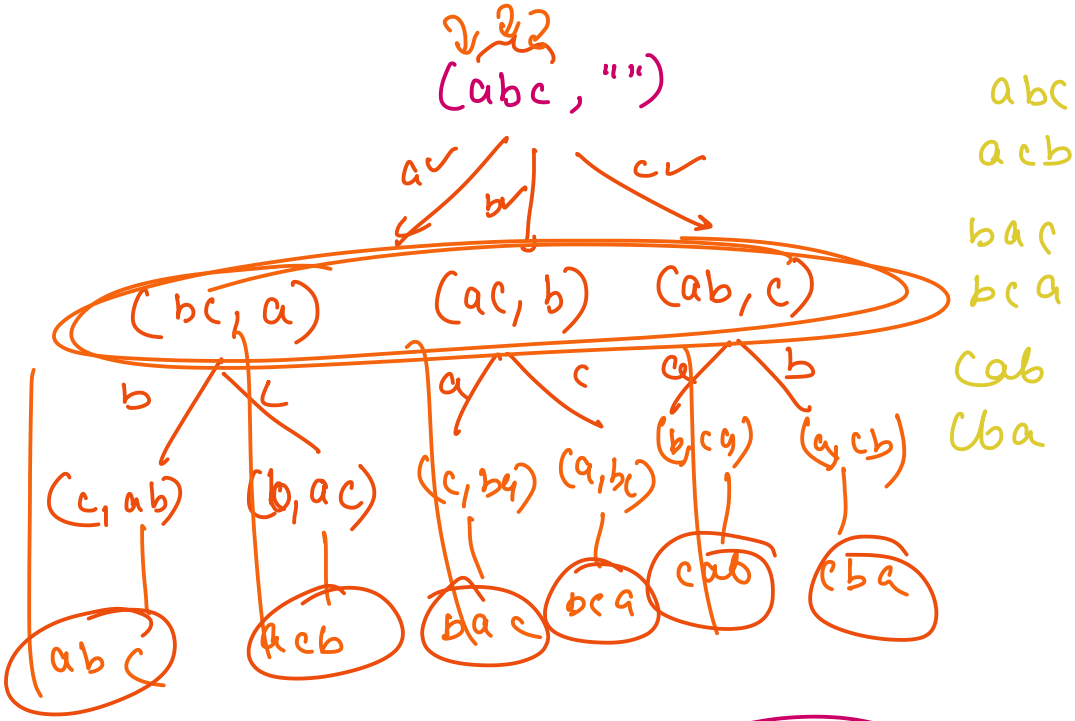
Q: abc → "" a

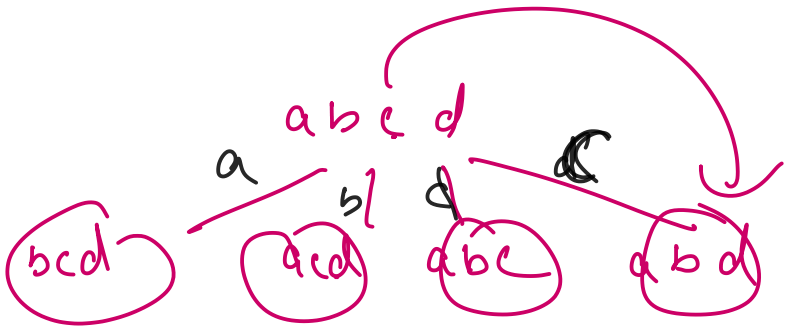












str.substring (0, 2) + str.substring(

ab ⟷ d

abd

$T(n) = T(n-1) * n$ → for loop

↓ recursion

$T(n) = n \times [2n + n! \times n + T(n-1)]$

(n → for loop)

$T(n) = n \times (T(n-1) + O(1))$