

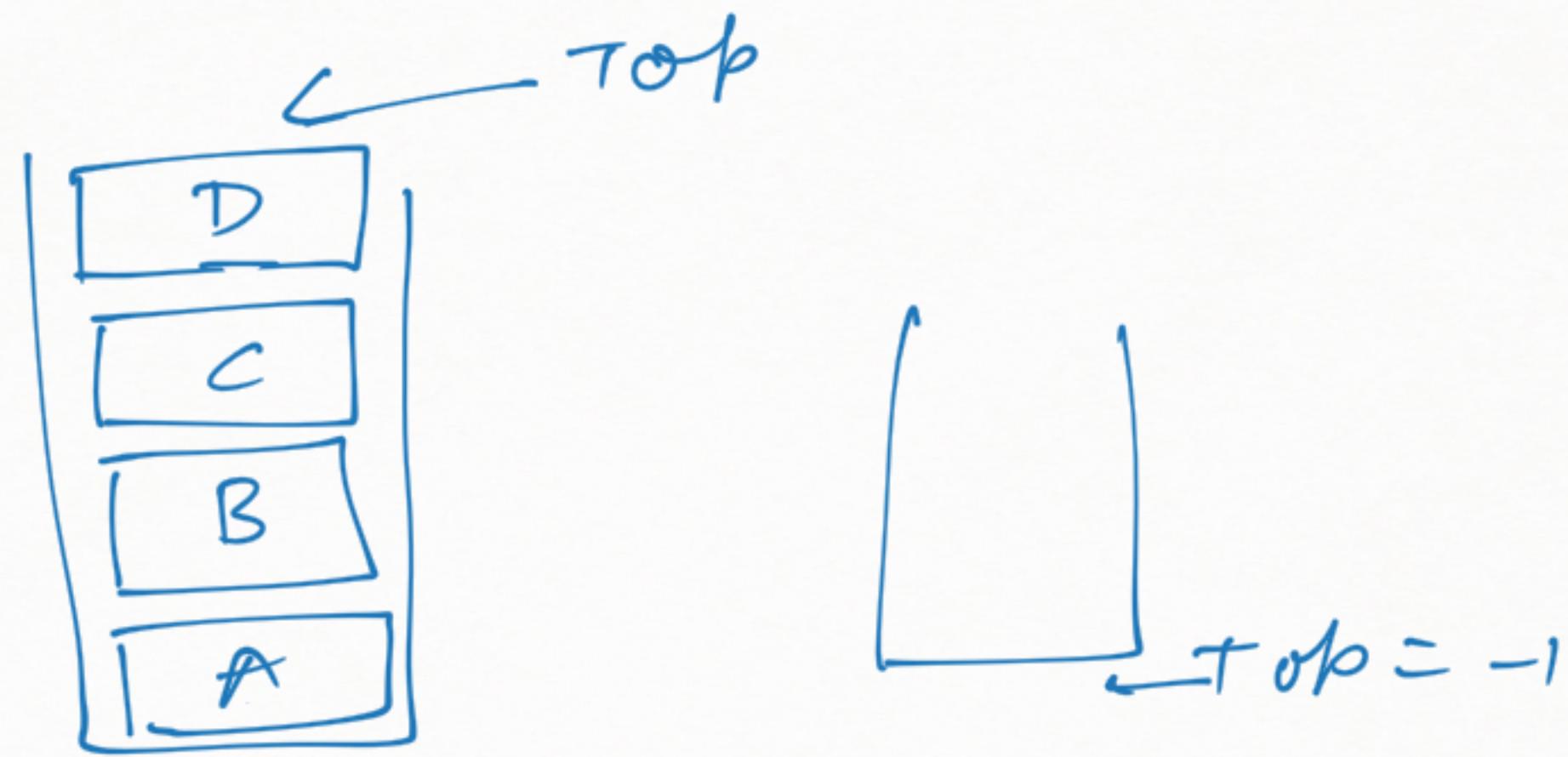
LIFO →

IN First

A  
B  
C  
D

OUT out

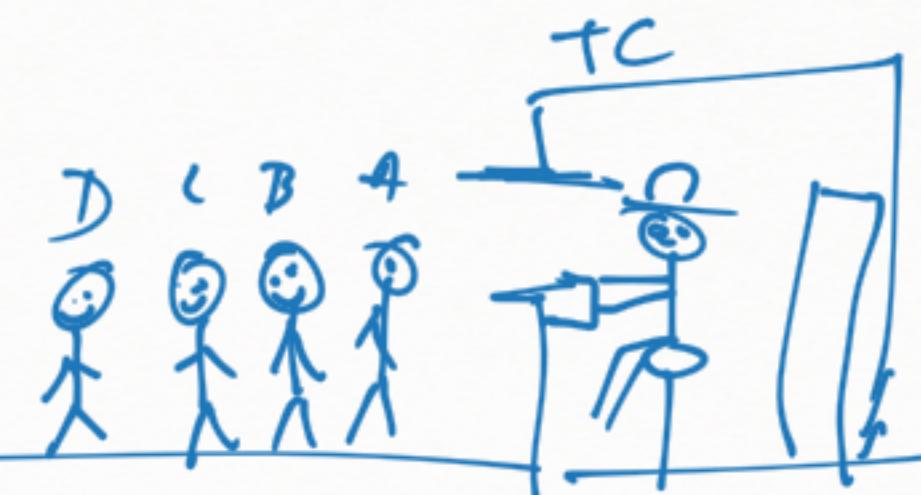
D  
C  
B  
A

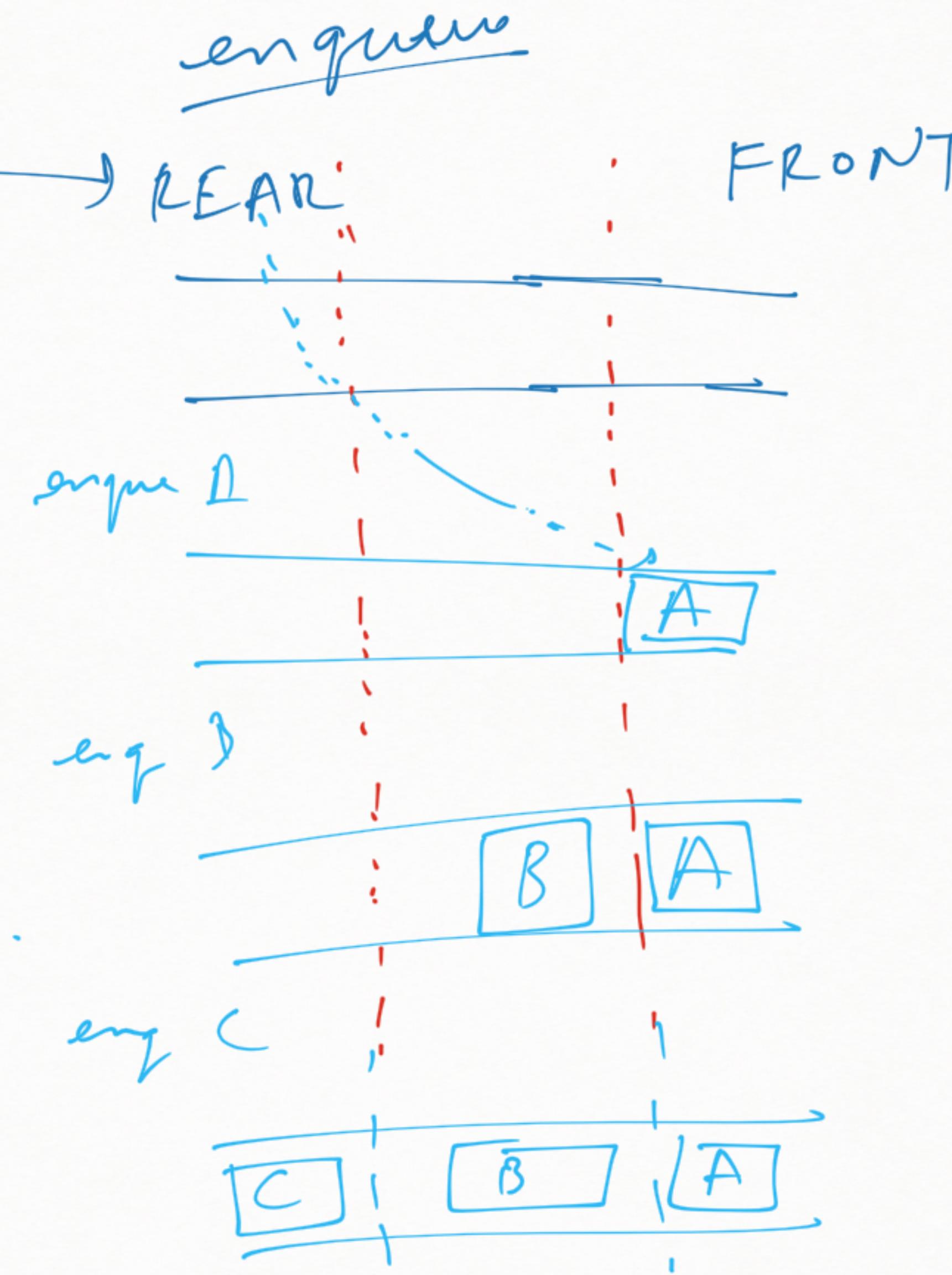
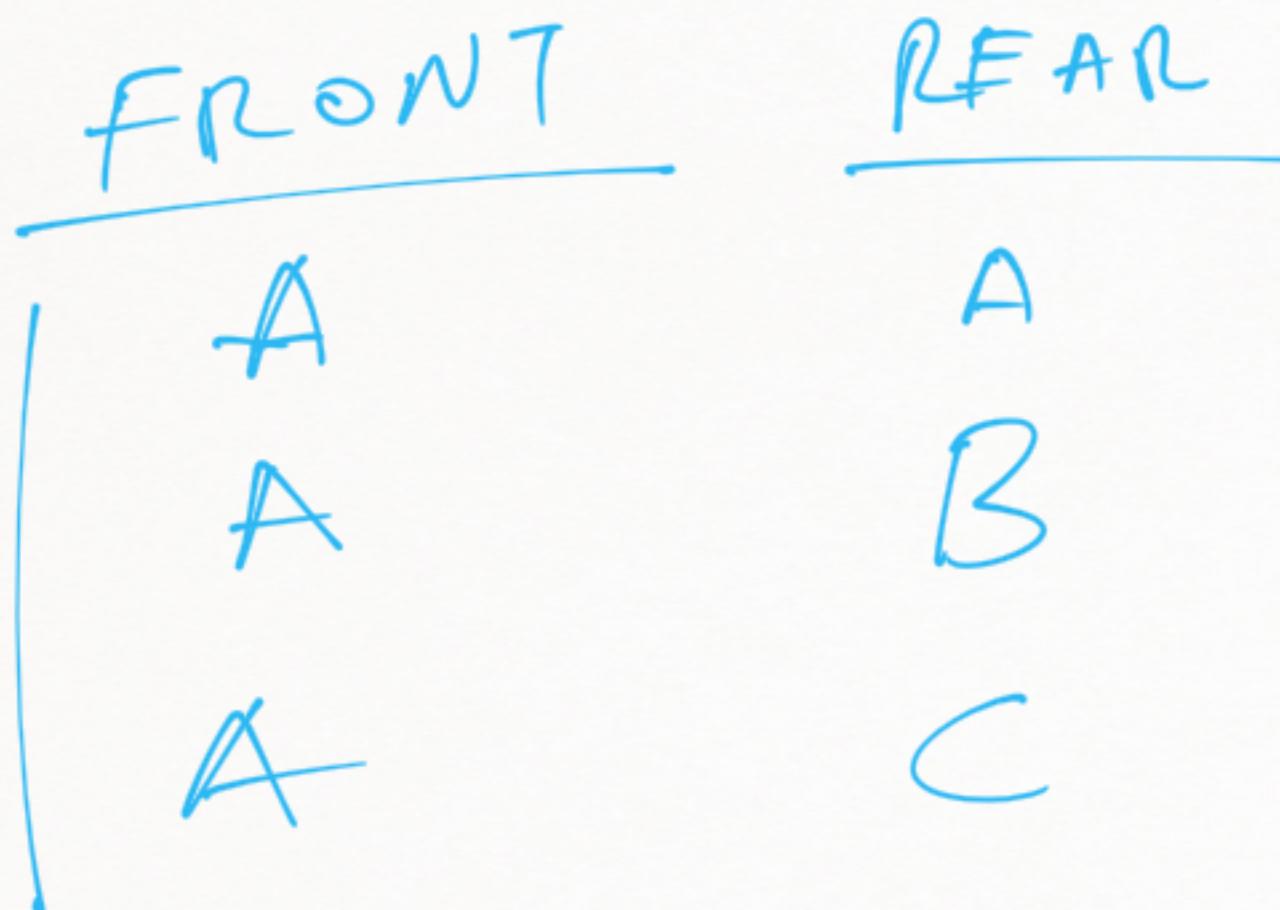
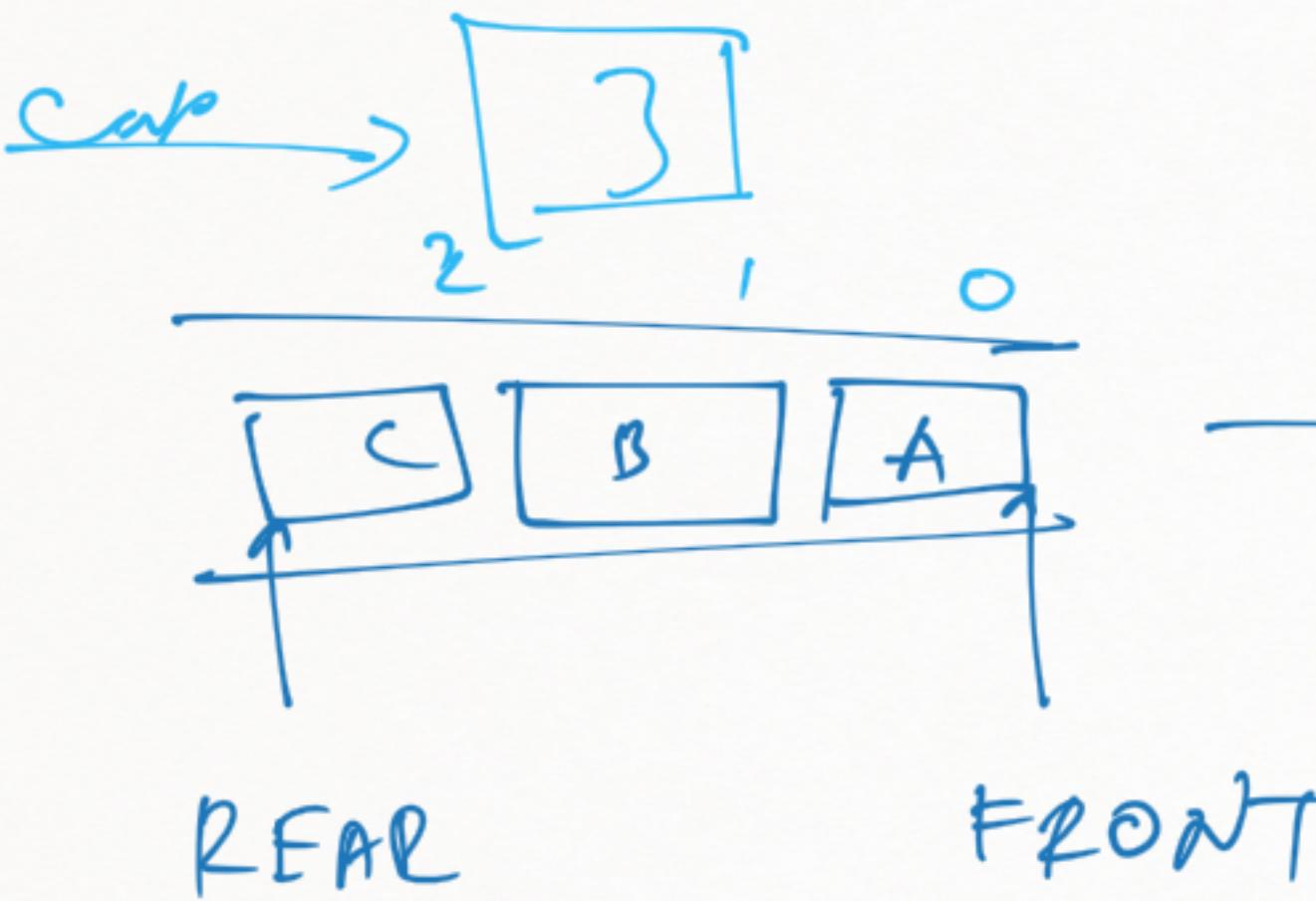


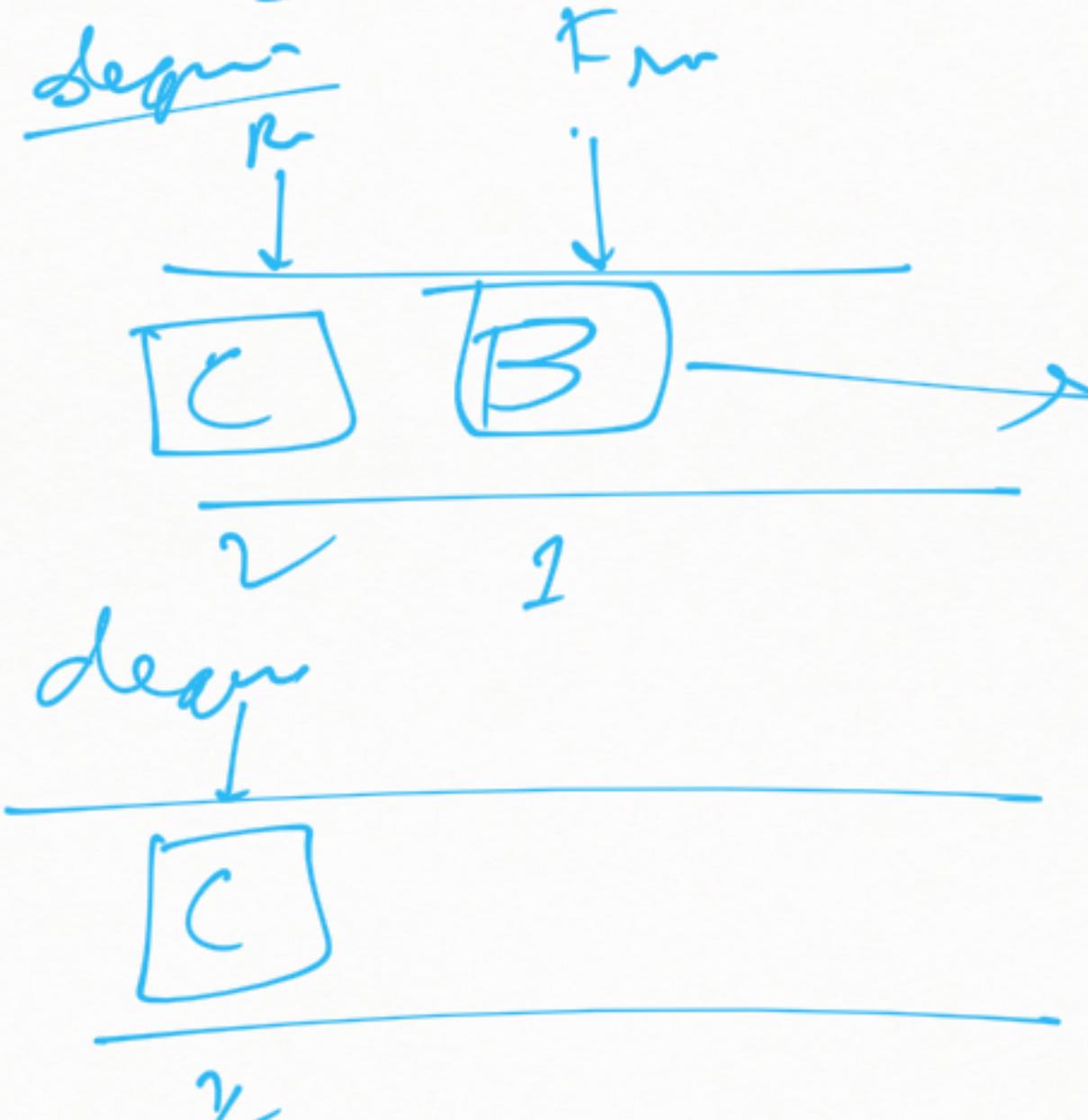
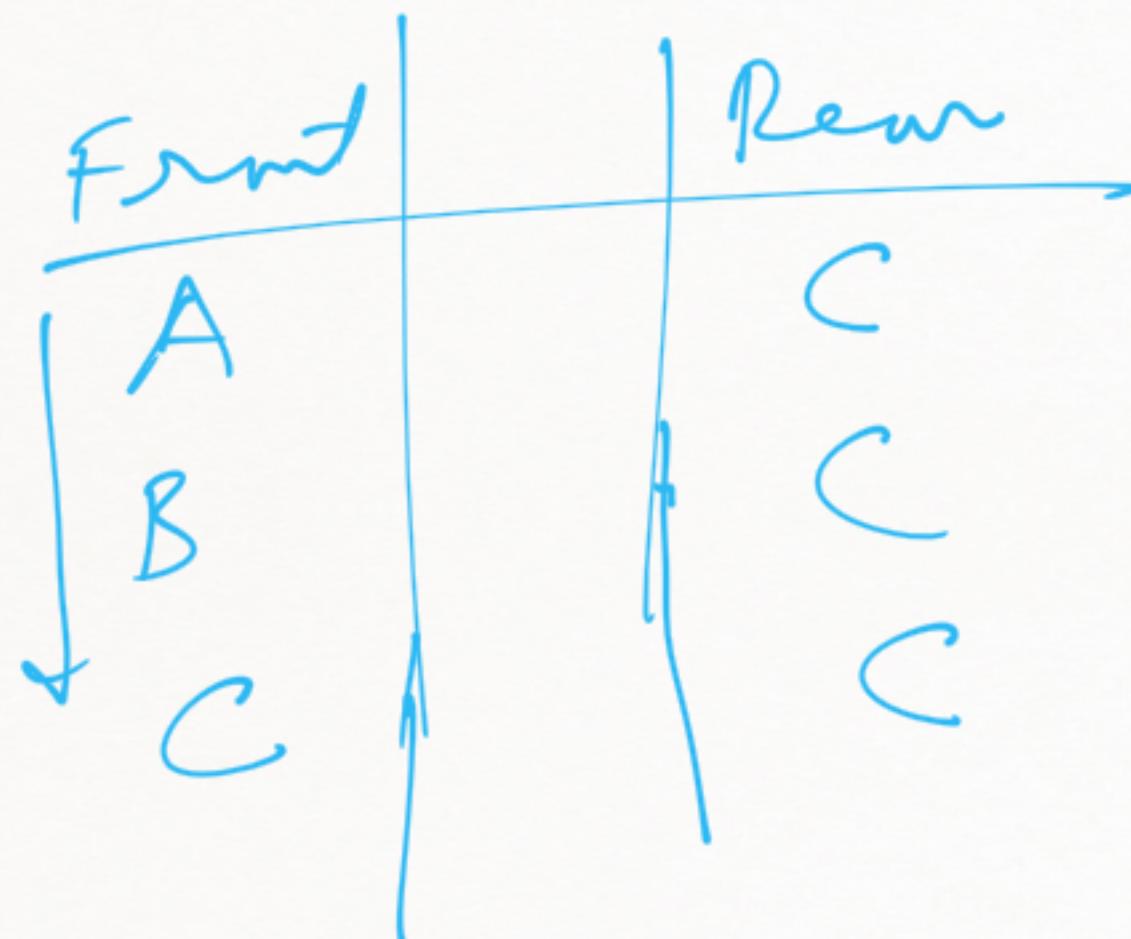
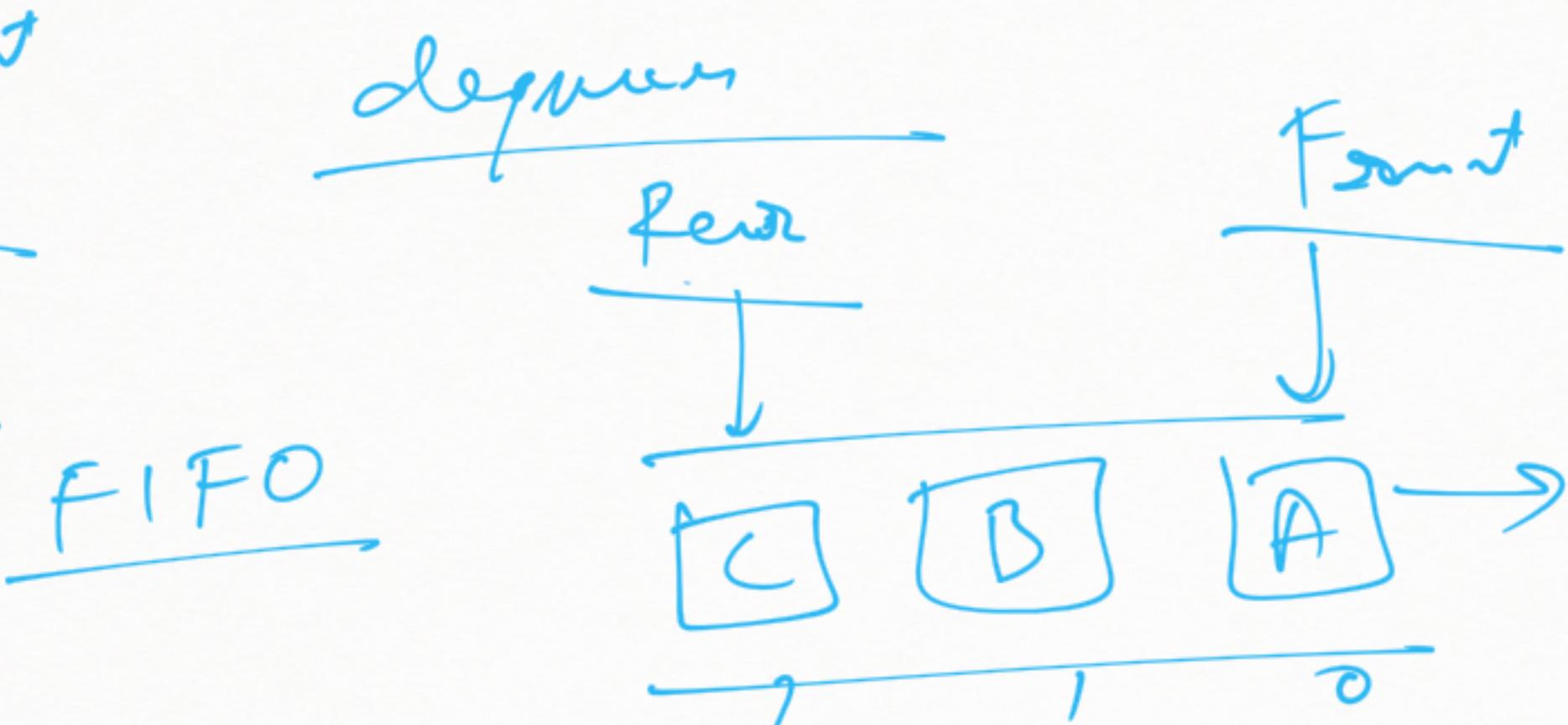
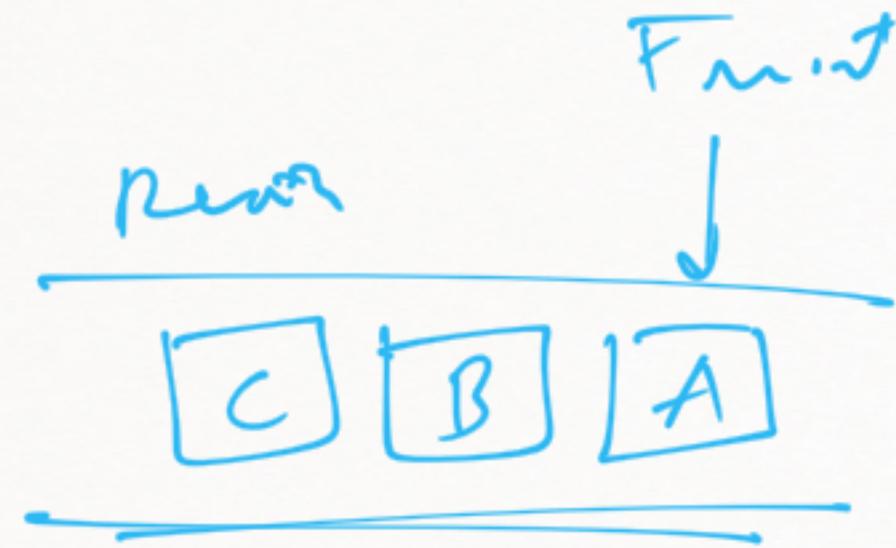
Queue

FIFO

$A \rightarrow B \rightarrow C \rightarrow D$





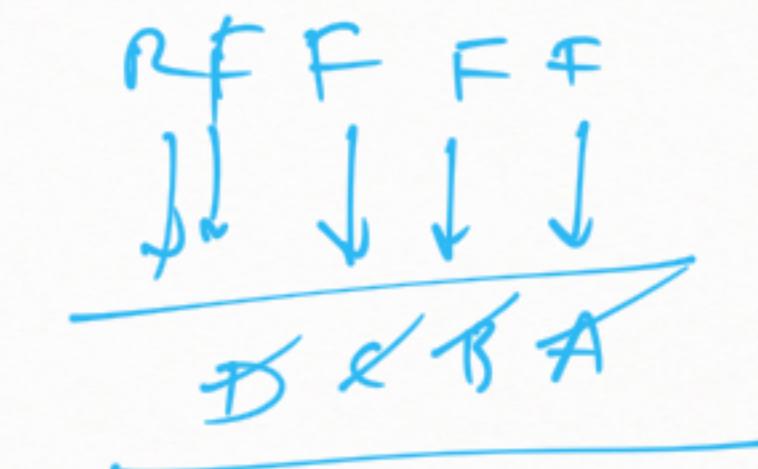
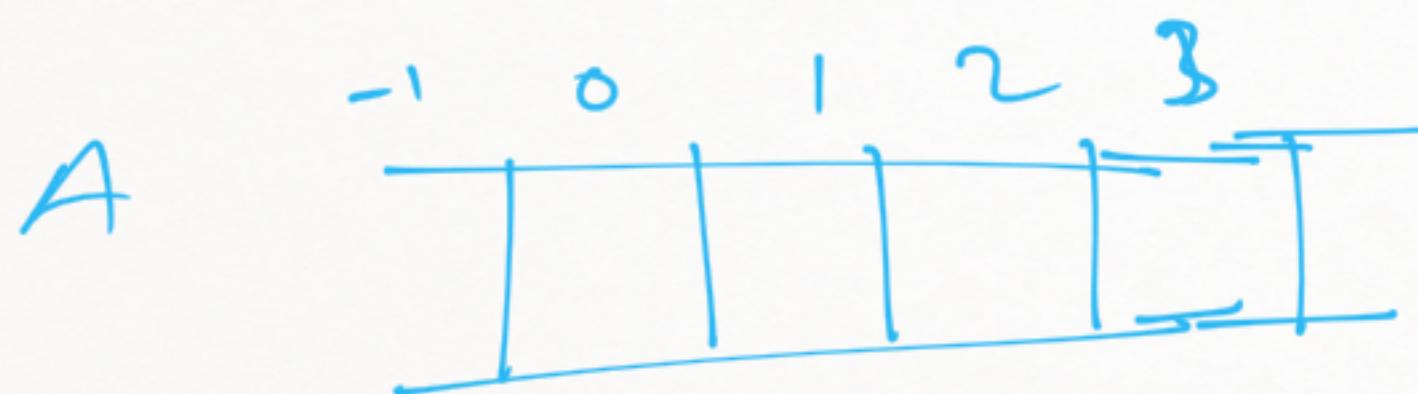
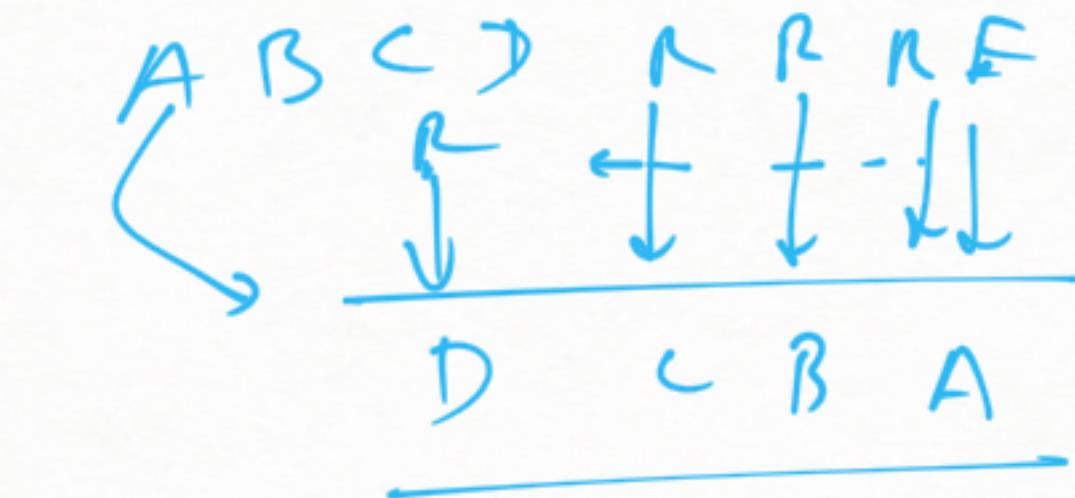


⑤

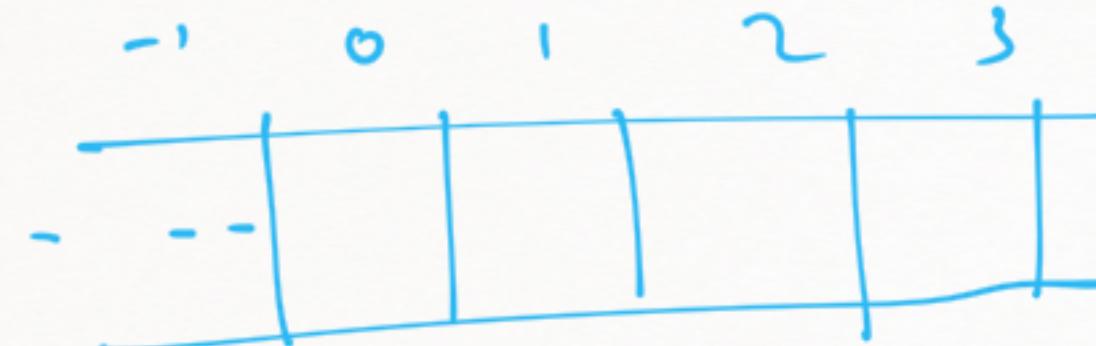
Stack using queue

Stack  $\rightarrow$  LIFO

queue  $\rightarrow$  FIFO



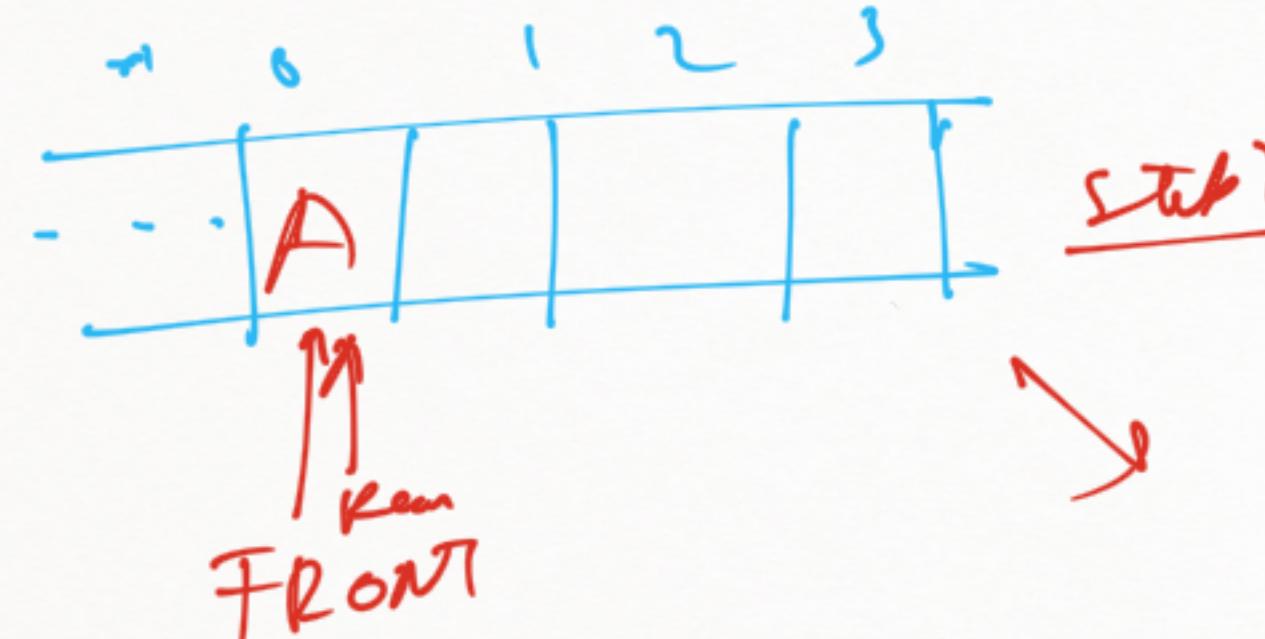
2



$\overrightarrow{AB}$   $\square CD$

degree from B and equal to A

23



1

2)

2) 9

2)

1 red  
FM

10

2)

3)

4

)

6)

refut  
↓↓

A---

$\beta$   $\alpha \beta \gamma$

3

A hand-drawn diagram illustrating the formation of an image by a lens. A horizontal line at the bottom represents the ground. Above it, a horizontal line represents the optical axis. A curved lens, drawn with a red marker, is positioned above the axis. Two parallel light rays from the left enter the lens from the top. The lens refracts the rays towards each other. The point where the rays converge is labeled 'f' (focal length). The image of a real object 'AB' is formed on the ground at a distance 'f' to the left of the lens. The image is inverted and labeled 'BA'. A curved arrow on the left indicates the direction of light rays.

A horizontal red line with a small red dot at its left end.

A red arrow pointing diagonally upwards and to the right, indicating a positive trend or flow.

C

A horizontal red line with a small break in the middle, followed by a vertical red line on the right side.

A B

new proved A B

$\alpha$

$A \beta C$

$\gamma$

$A \gamma C$

$\delta$

$A \delta C$

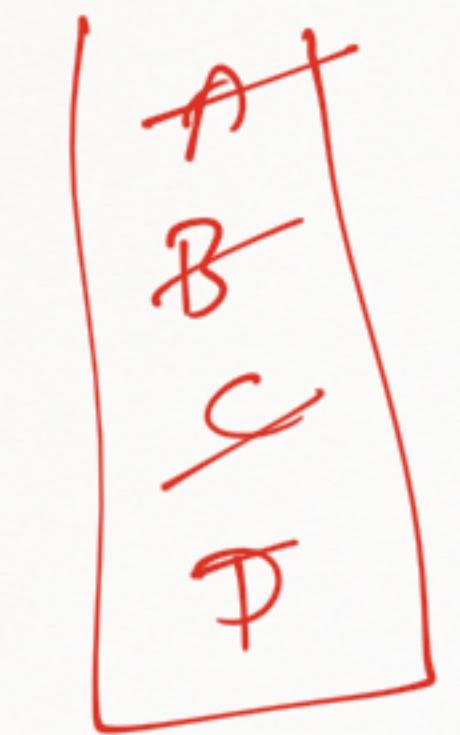
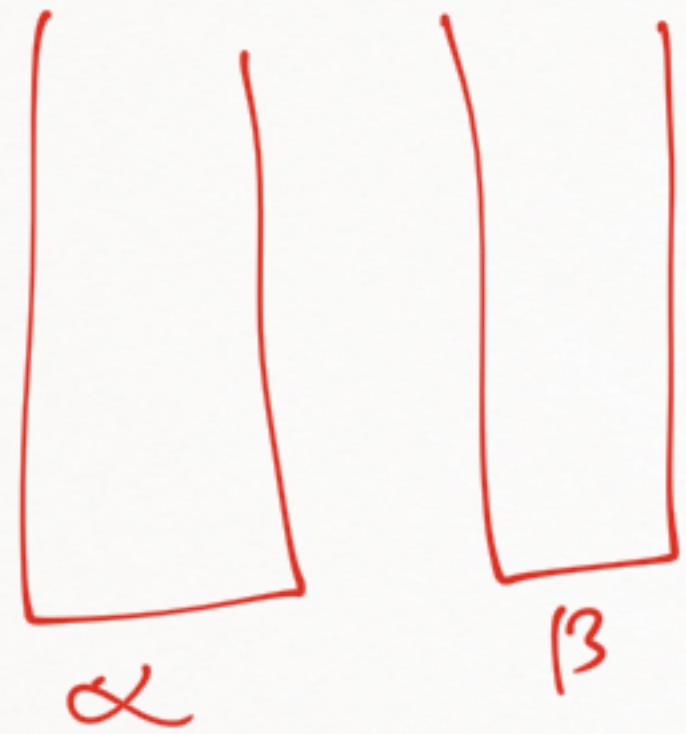
$\epsilon$

got      A B C D

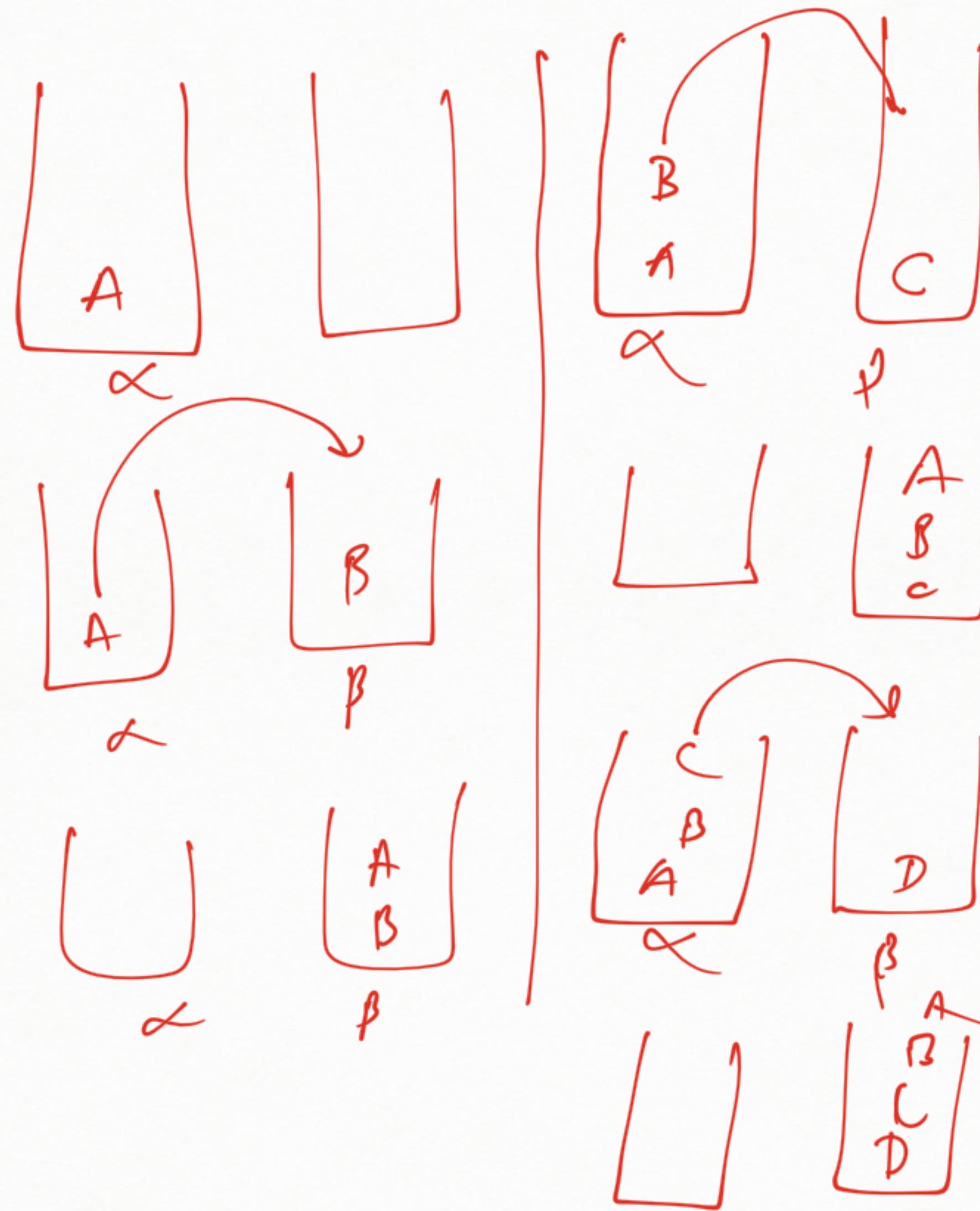
entered    D C B A

## Queue using stack

D-C-B-A



D C B A



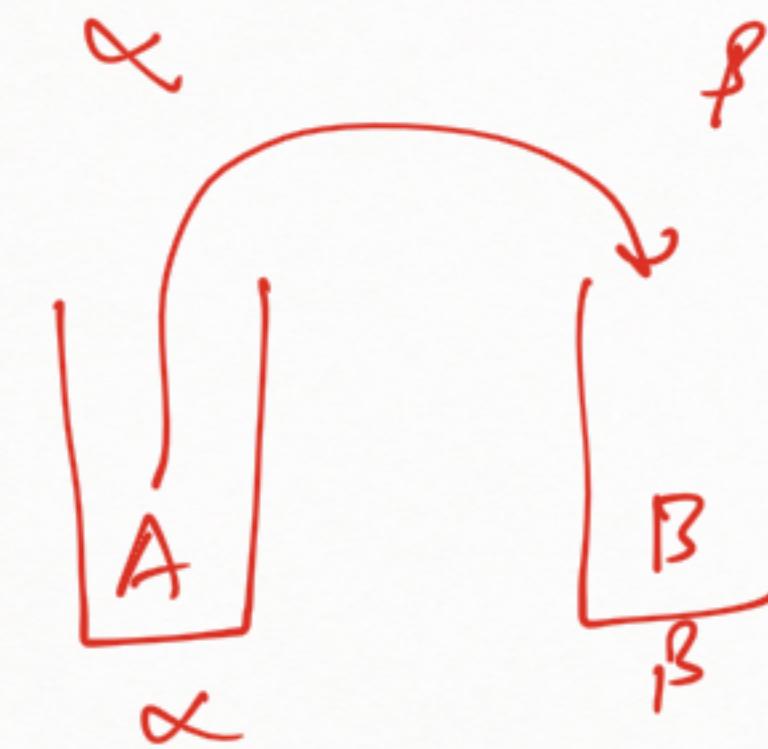
$D \leftarrow C \leftarrow B \leftarrow A$



Step 1



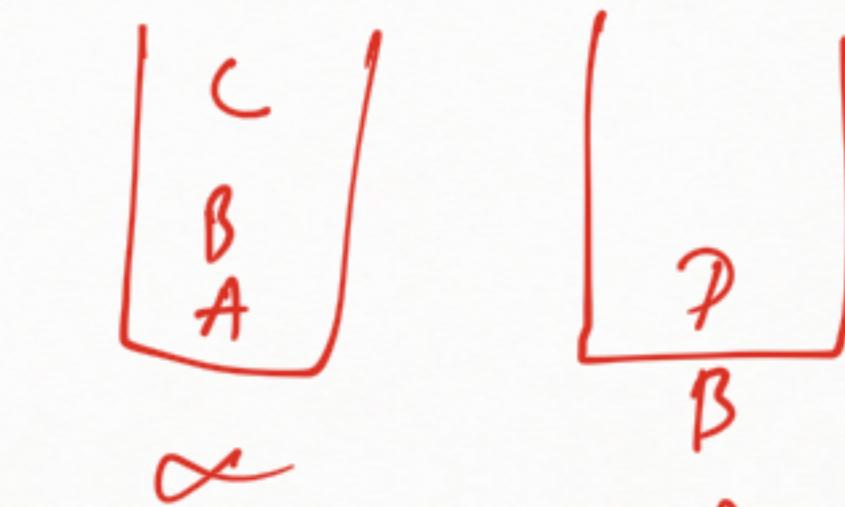
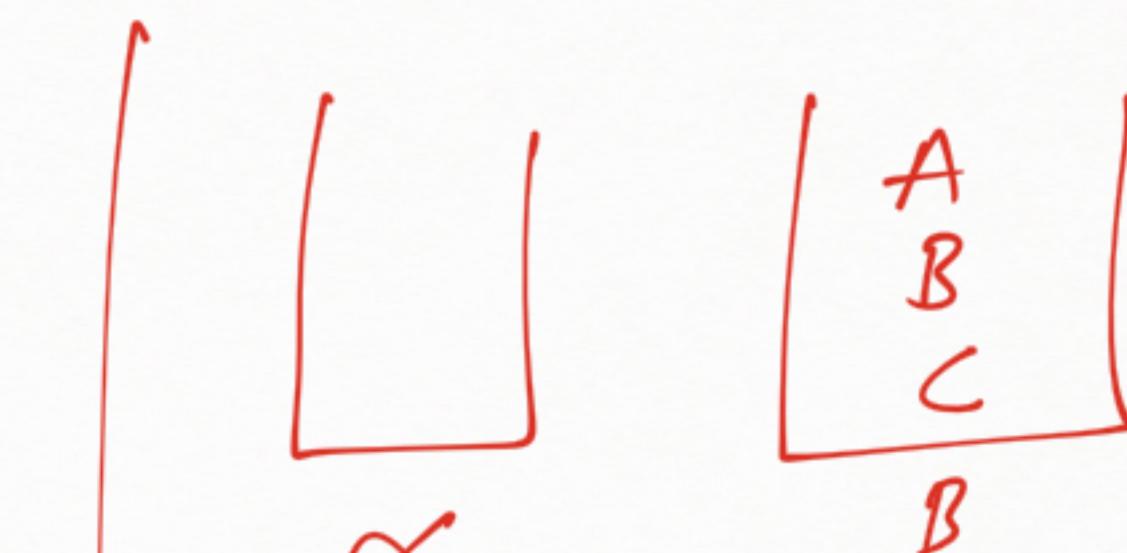
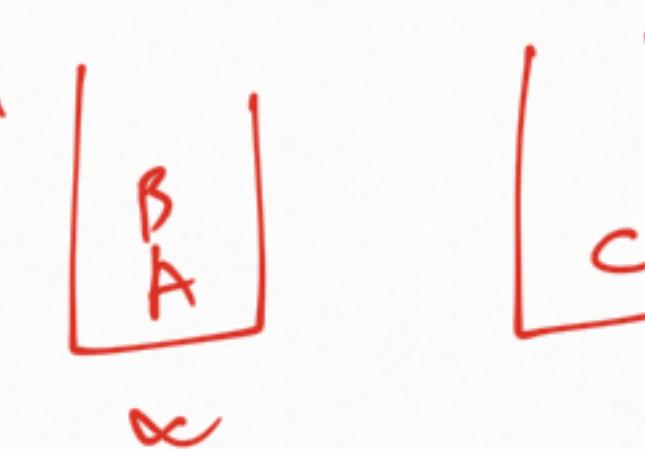
Step 2



Step 3



Step 4



$A \rightarrow B \rightarrow C \rightarrow D$

Code 1

~~T(n)~~  
~~T(0)~~

## TIME COMPLEXITY

Space ↓      Time ↓



$A > B$   
 $A < B$   
 $A = B$

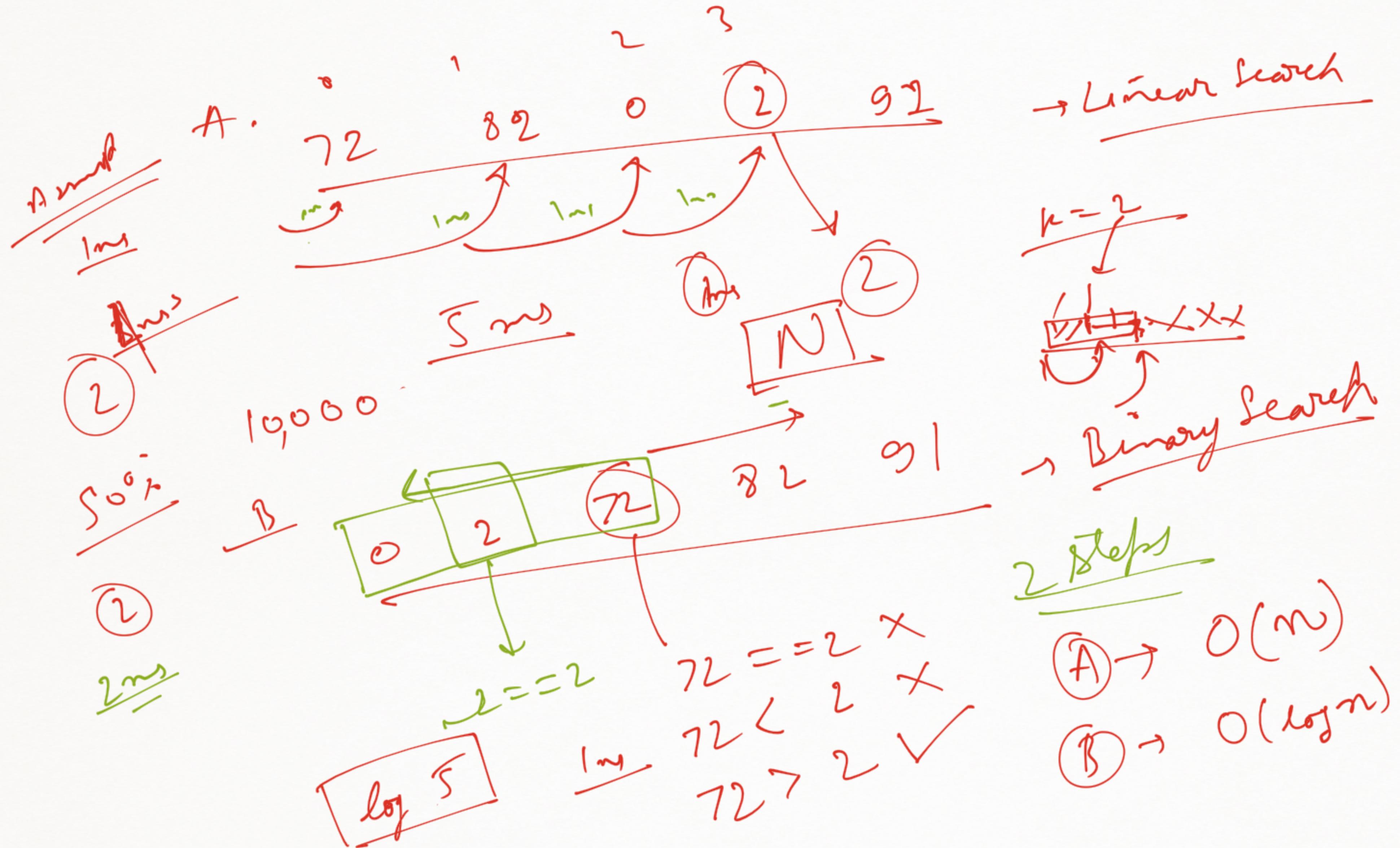
Code 2  
O(n)  
Time

efficiency

worst

Asymptotic notation

- O Big oh Notation ✓
- Ω Big Omega
- Θ N notation



→ Linear Search

$$k = 2$$



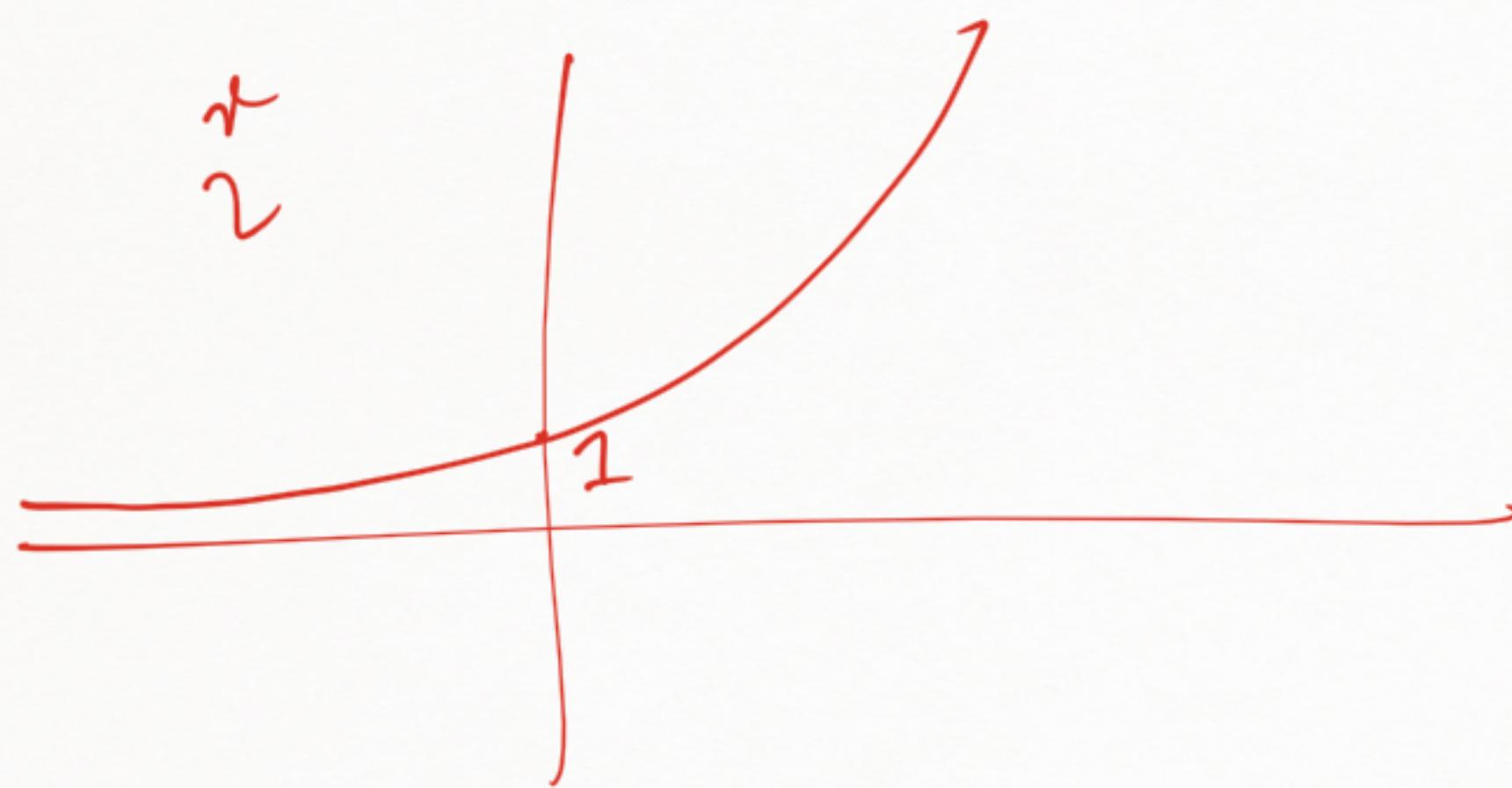
→ Binary Search

2 Steps

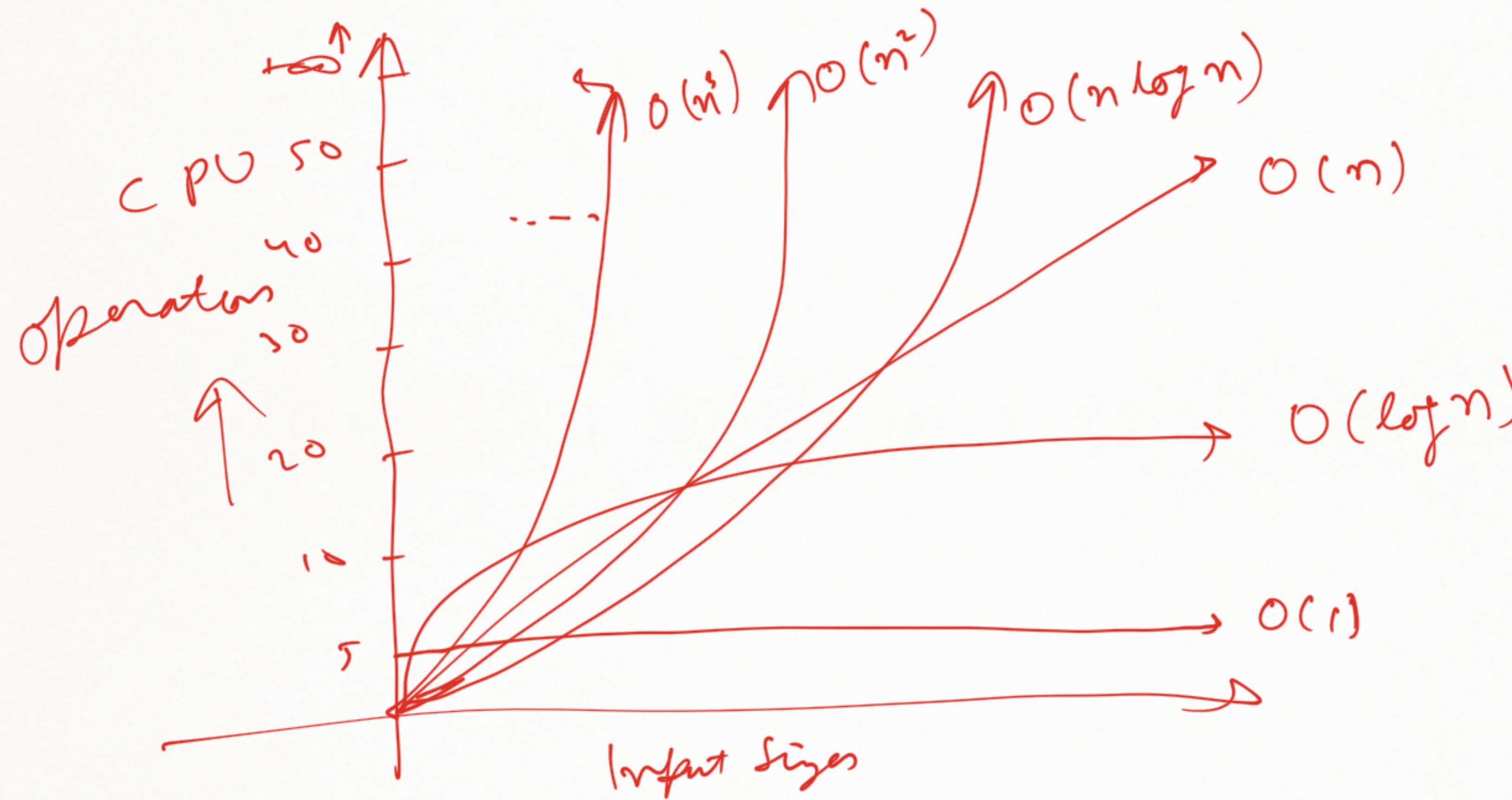
$$\textcircled{A} \rightarrow O(n)$$

$$\textcircled{B} \rightarrow O(\log n)$$

$$2 = 2^n$$
$$\log 2 = n$$



$$\underline{T(n) = O(4n^2 + 7n + 3)}$$



$O(1) \longrightarrow$

$O(n)$

$a = 5$

$b > c \quad O(1)$

$s_1$

$s_2$

:

$s_N$

$$total = time(s_1) + time(s_2) + \dots + time(s_N)$$

$$T(n) = f(s_1) + f(s_2) + f(s_3) + \dots + f(s_n);$$

$$\overbrace{\phantom{f(s_1) + f(s_2) + f(s_3) + \dots + f(s_n);}}$$

## Sum of 3 numbers

def fun (a, b, c) :

$$s_a = a \rightarrow O(1)$$

$$s_b = b \rightarrow O(1)$$

$$s_c = c \rightarrow O(1)$$

$$s = s_a + s_b + s_c \rightarrow O(1)$$

$$\text{return } s \rightarrow O(1)$$

$$\begin{aligned} T(n) &= O(1) + O(1) + \dots \\ &= \underbrace{O(1)}_{\text{loop}} \approx O(1) \end{aligned}$$

for  $i$  in range( $\text{length}^n$ ):

$s_1$

$s_2$

$$T(n) = m \times [+(s_1) + +(s_2) + \dots]$$
$$= O(n)$$

for i in range(m) :  
    for j in range(m) :  
        s1  
        s2

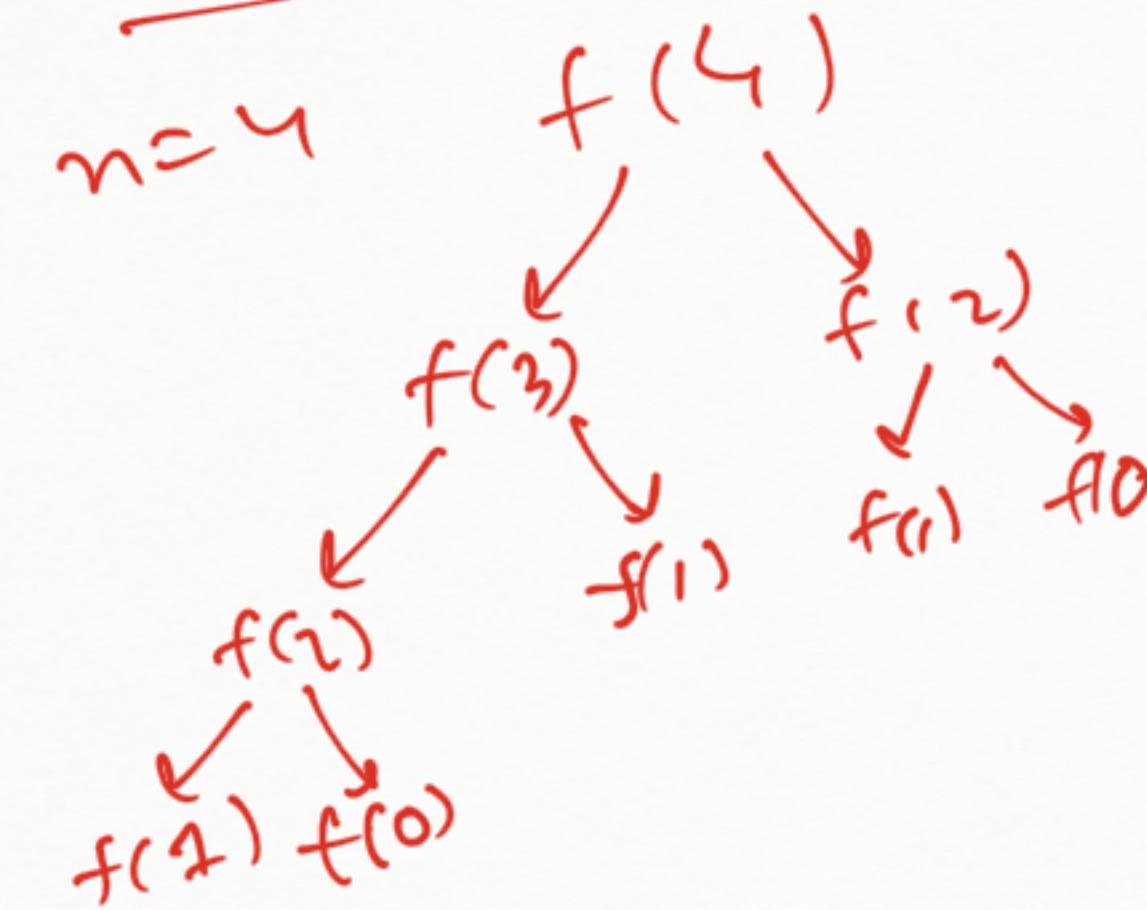
$$\begin{aligned}T(n) &= n \times [t(s_0) + \overbrace{m \times [t(s_1) + t(s_2)]}^{\text{inner loop}}] \\&= n t(s_0) + \underbrace{m \times m \times [t(s_1) + t(s_2)]}_{\text{outer loop}} \\&= O(n \times m) \approx O(n^2)\end{aligned}$$

`def func(n):`

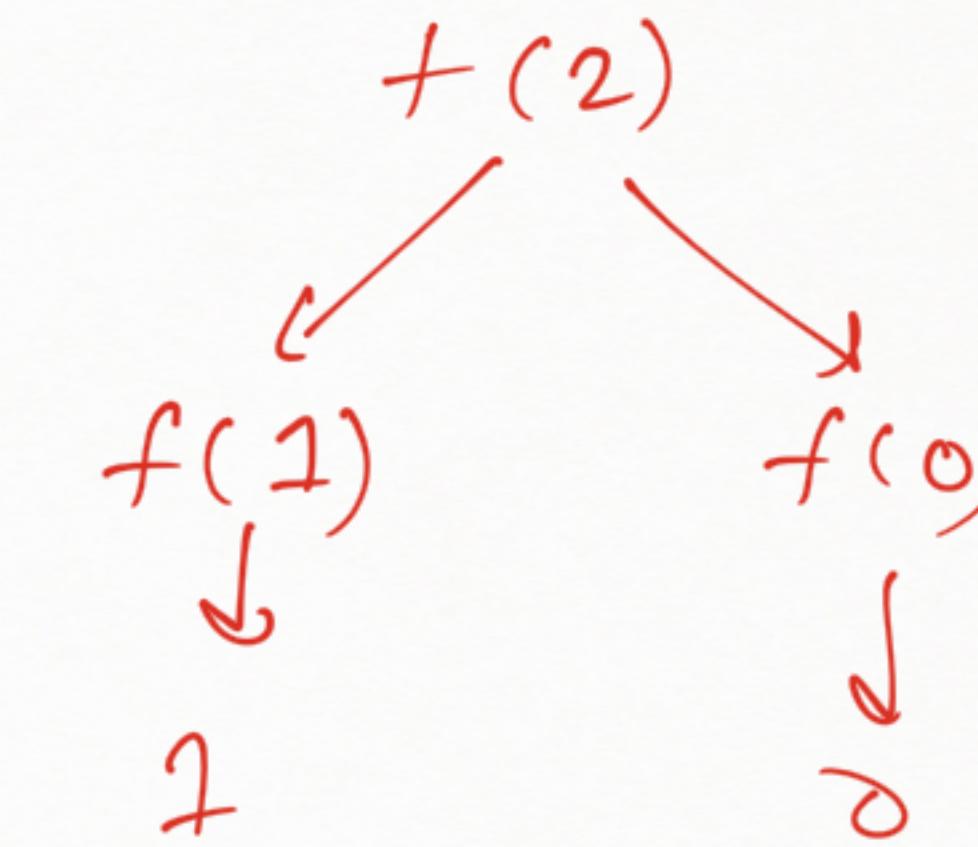
`if n < 0:`  
 `return 0`

`if n < 2: → O(2^n)`  
 `return n`

`return func(n-1) + func(n-2)`

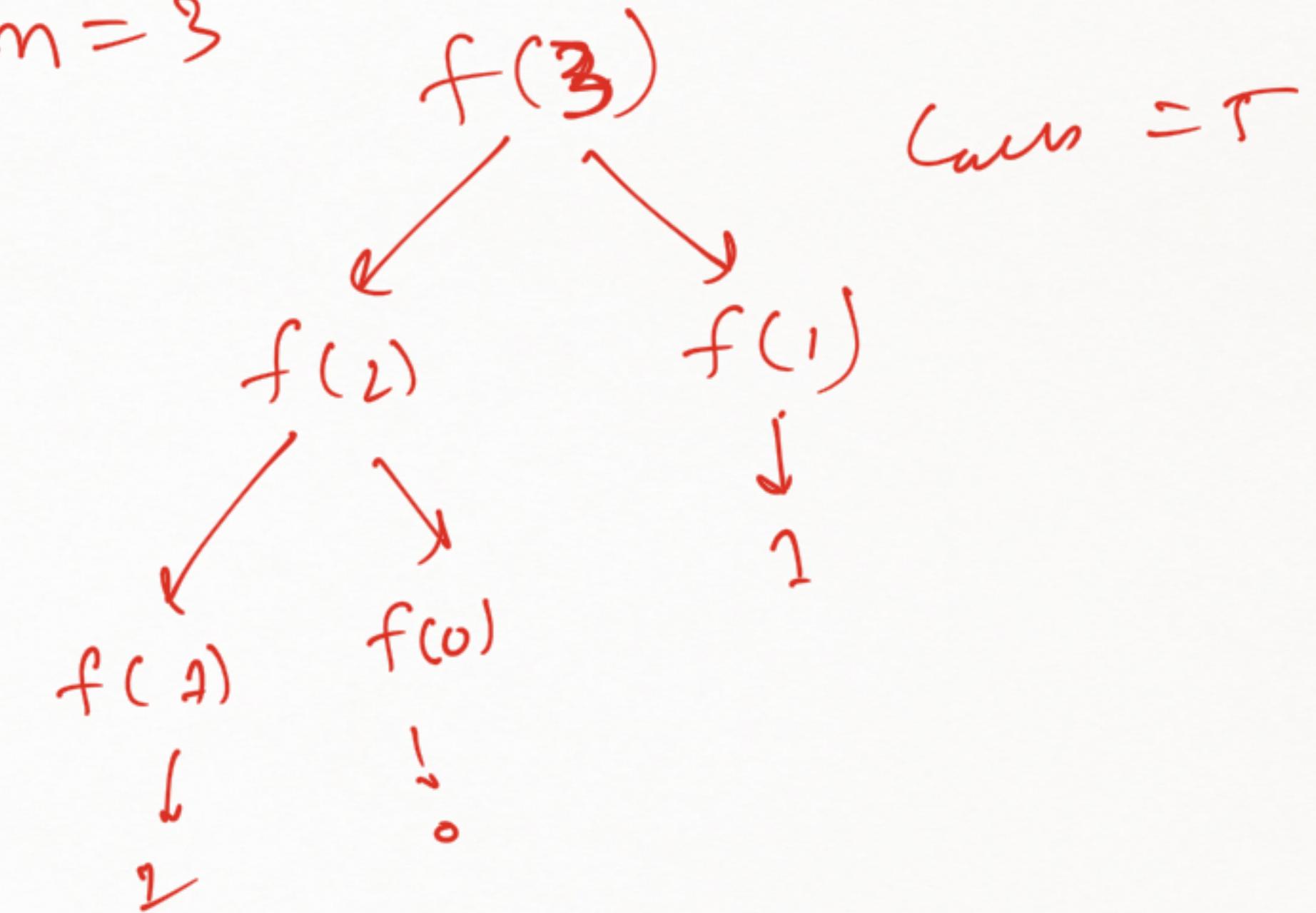


$n=2$



calls = 3

$n=3$



calls = 5