```
public static void fName(int a, char c)
{
    SOP("Hello");
    SOP(a);
    SOP(c);
    return;
}
```

main

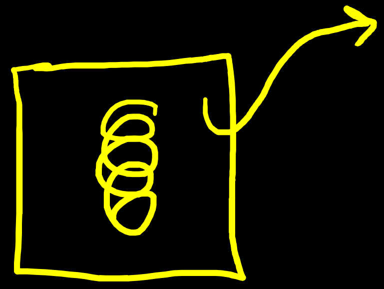
fName()|||

fName

a, c

return a;

i/p

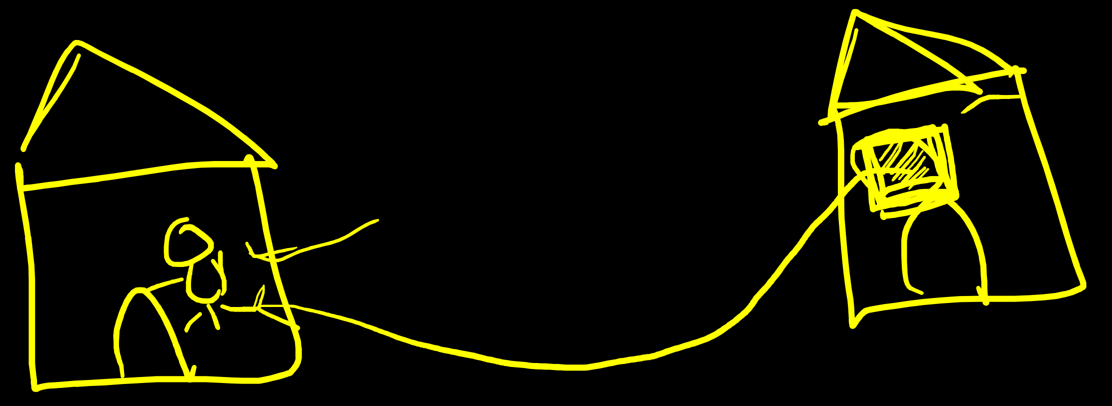


hello

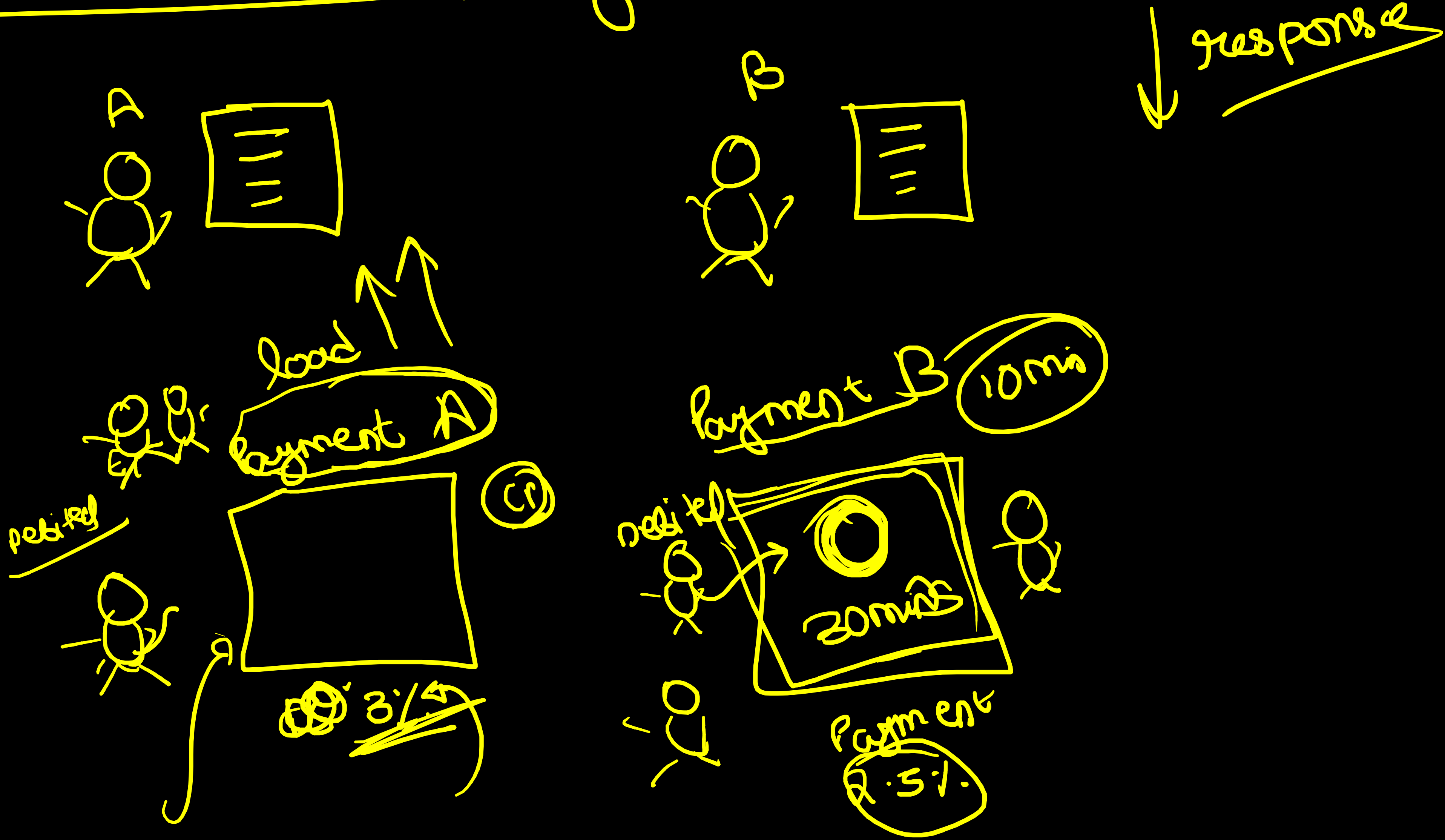
5

'A'

```
main()
    b = 5
{
    fName(b, 'A');
    SOP(a);
}
```



Time and Space Complexity



Space

program

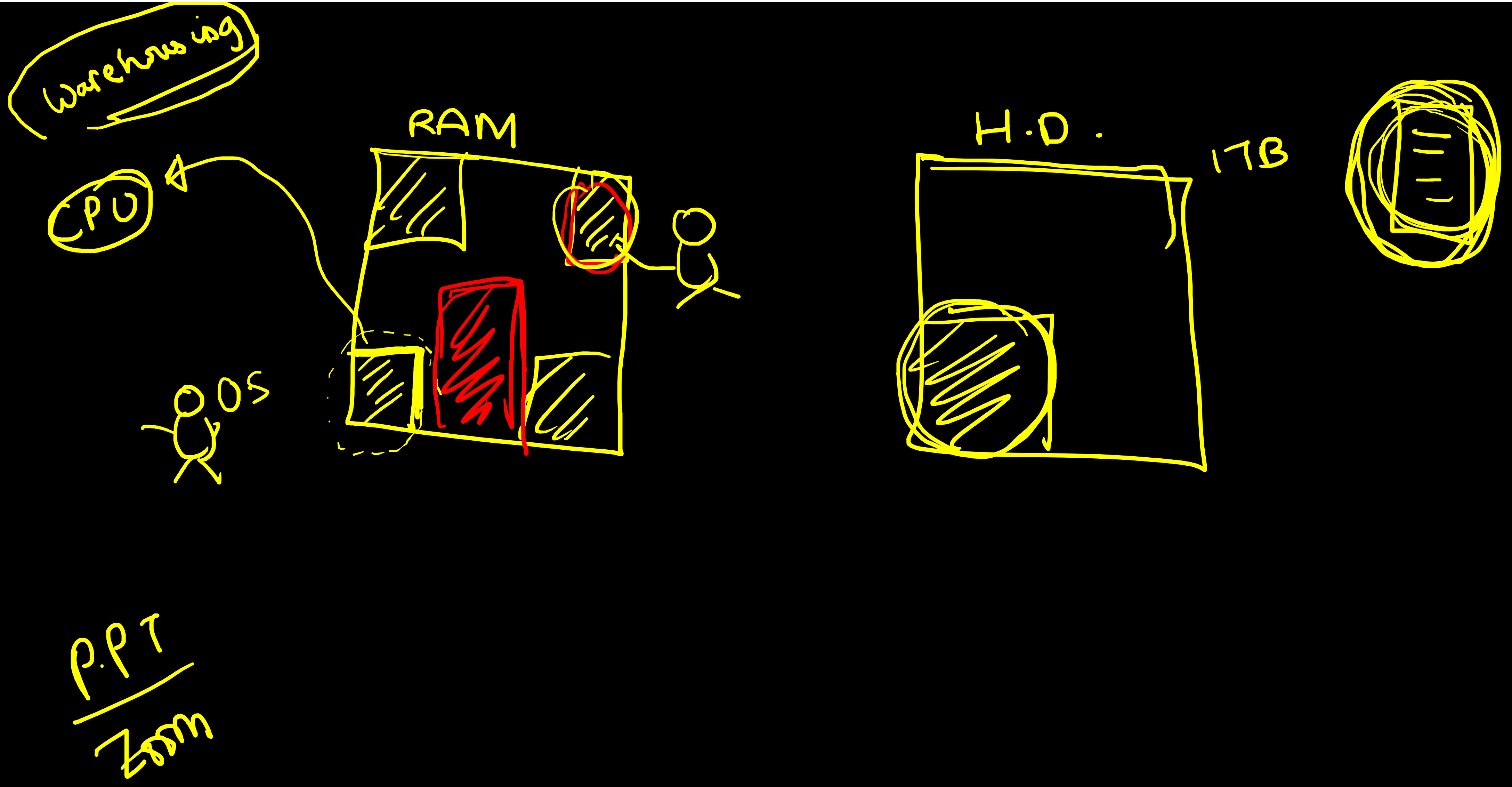
RAM

Application

5MB

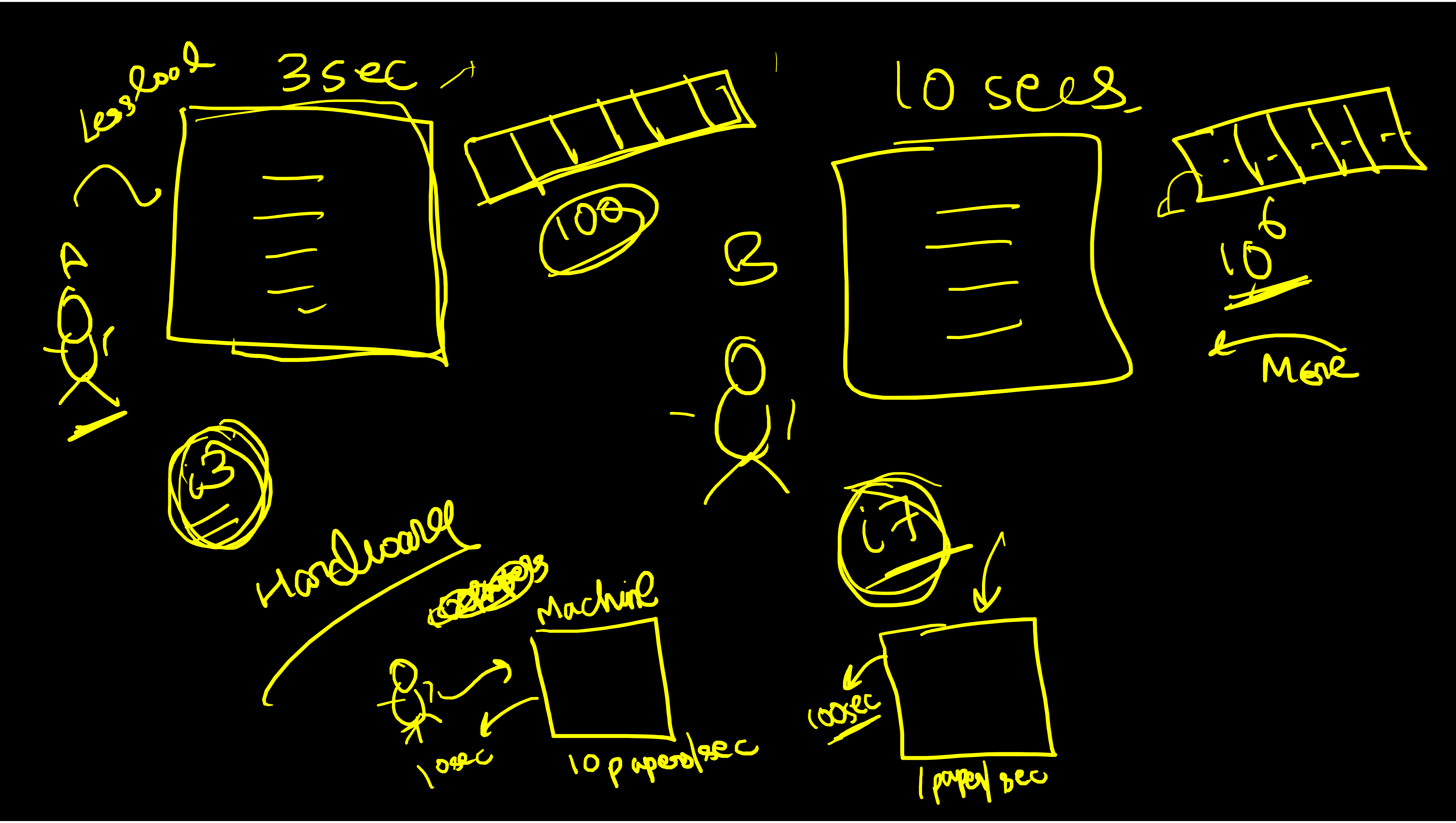
Application

1GB


Warehousing
CPU
RAM
OS
H.D.
1TB
P.PT
Zoom

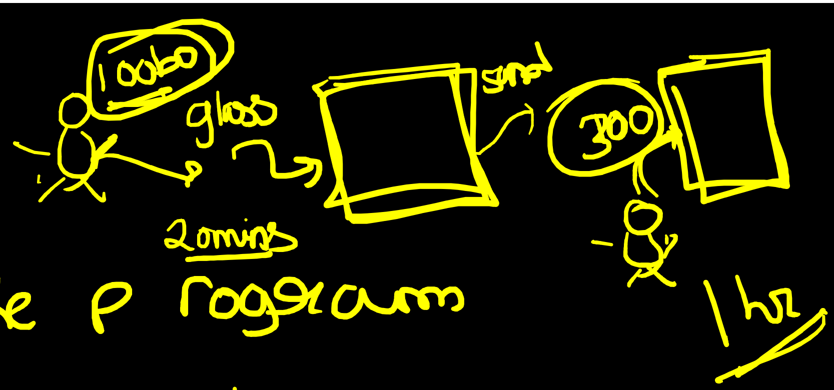
Time & Space

Time / Space


Less load
3 sec
100
A
i3
B
10 secs
10
More
i7
Hardware
Machine
10 sec
10 papers/sec
100 sec
1 paper/sec

Time Complexity

→ Time taken by the program as a fn of i/p size.



$$T = f(n)$$

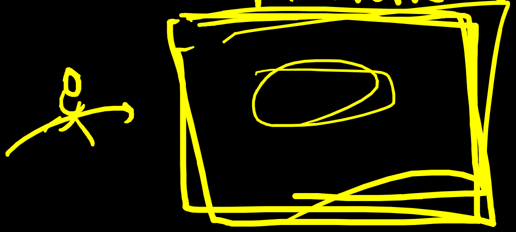
A

Bottles

100 → ~~1000~~ 500 min

B 1000 → 5000 min

Machine A



$T \propto n$

$100 \rightarrow 500\text{sec}$

$1000 \rightarrow 5000\text{sec}$

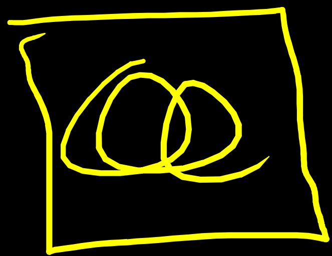
$\uparrow T \propto n \uparrow$

n

10^5

$n^2 > n$

Machine B



n^2

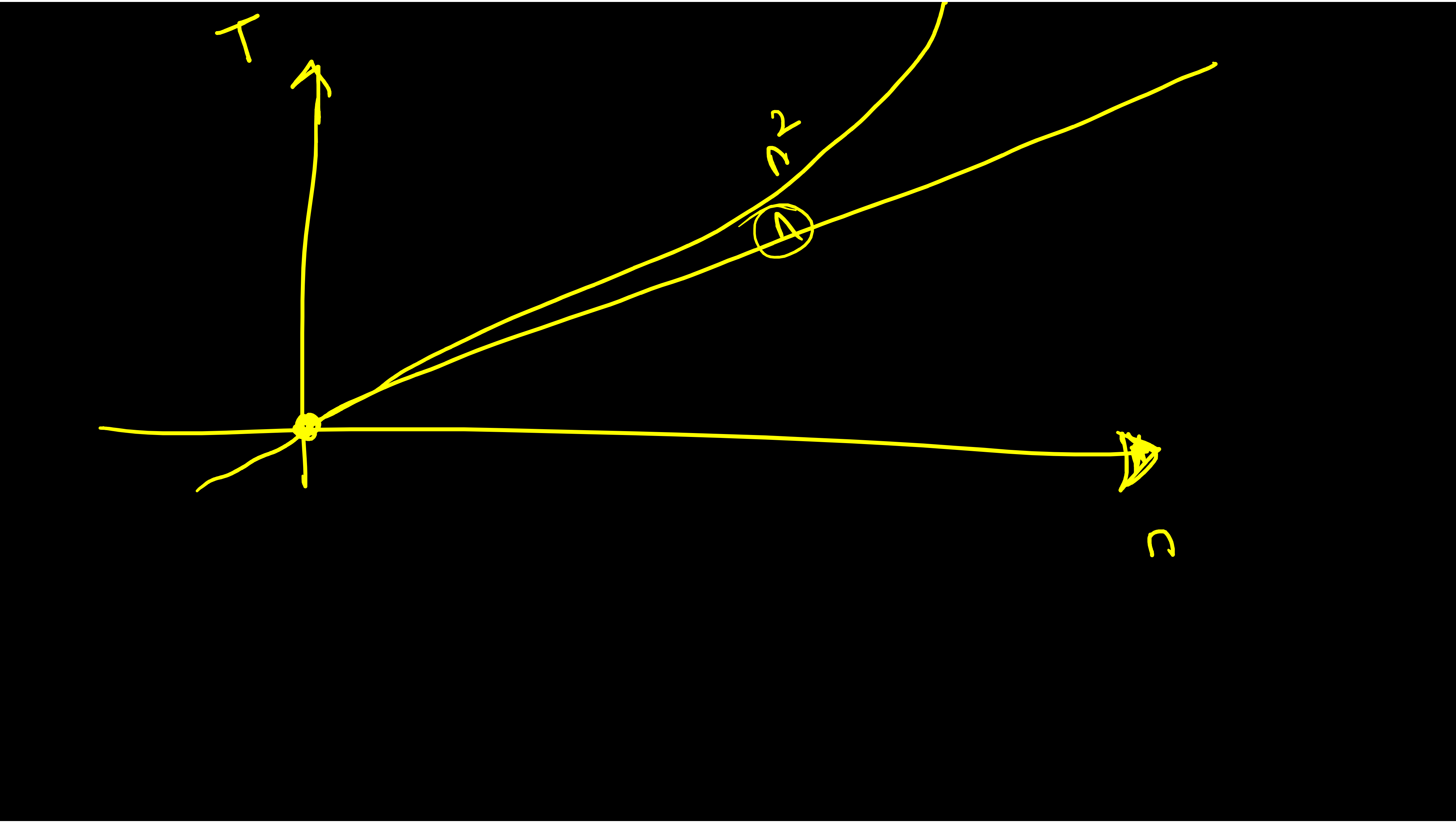
$5 \rightarrow 25\text{ secs}$

$10 \rightarrow 100\text{sec}$

$15 \rightarrow 225\text{s}$

$T \propto n^2$

10^{10}


T
2
1
n

int n ; ✗

int a = 0 ; ✓

n = 10^5

for (int i = 0 ; i < n ; i++)
{ a = a+1; }

100 times

10^5

n

0

T ∝ n

```
int n = sc.nextInt();
int a = 0;
for(int i=0; i<n; i++)
{
    a = a+1;
}
```

0 times

n ↑

n times

$T \propto n$

```
int n;
int a = 0;
for (int i = 1; i <= n; i++)
{ for (int j = 1; j <= n; j++)
  { sop("Hello"); }
}
```

n = 3 → 9

n = 4 → 16

$T \propto n^2$

i = 1 → n times

i = 2 → n times

i = 3 → n times

i = 4 → n times

no

i = n → n times

$\underbrace{n + n + n + n \ldots}_{n \text{ times}} = n^2$

for(i=0; i<m; i++)
{ for(j=0; j<n; j++)
{
sop("Hi")
}
}

Loops

if else

i = 0 → n times
i = 1 → n times
i = 2 → n times
.
.
.
i = m-1 → n times

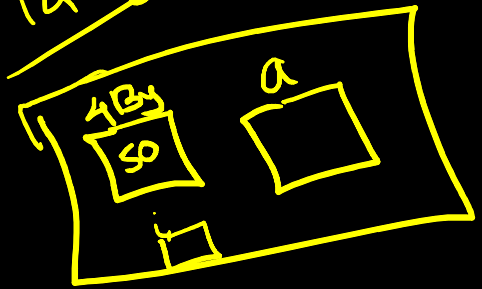
$\underbrace{n + n + n + n + \ldots}_{m\ \text{times}}$

m*n

Space -> f(n)

50

12 Bytes



int a

```
int n;
int a = 0;
for(int i=0; i<n; i++)
{a=a+1;}
```

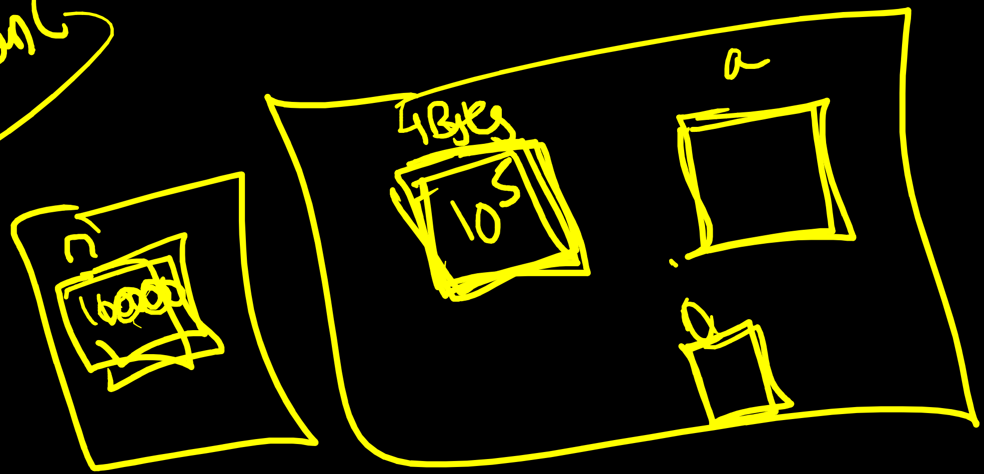
$2^{31}-1$

10^5

n=10000

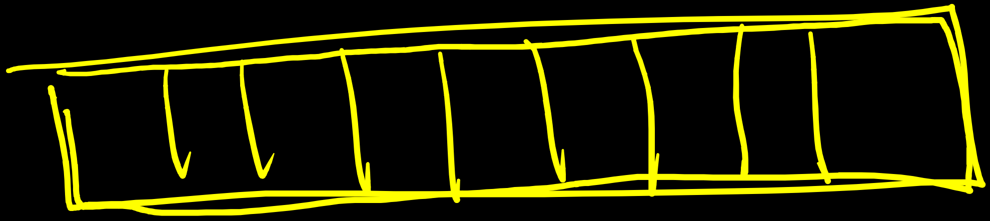
n=100

Space is Constant

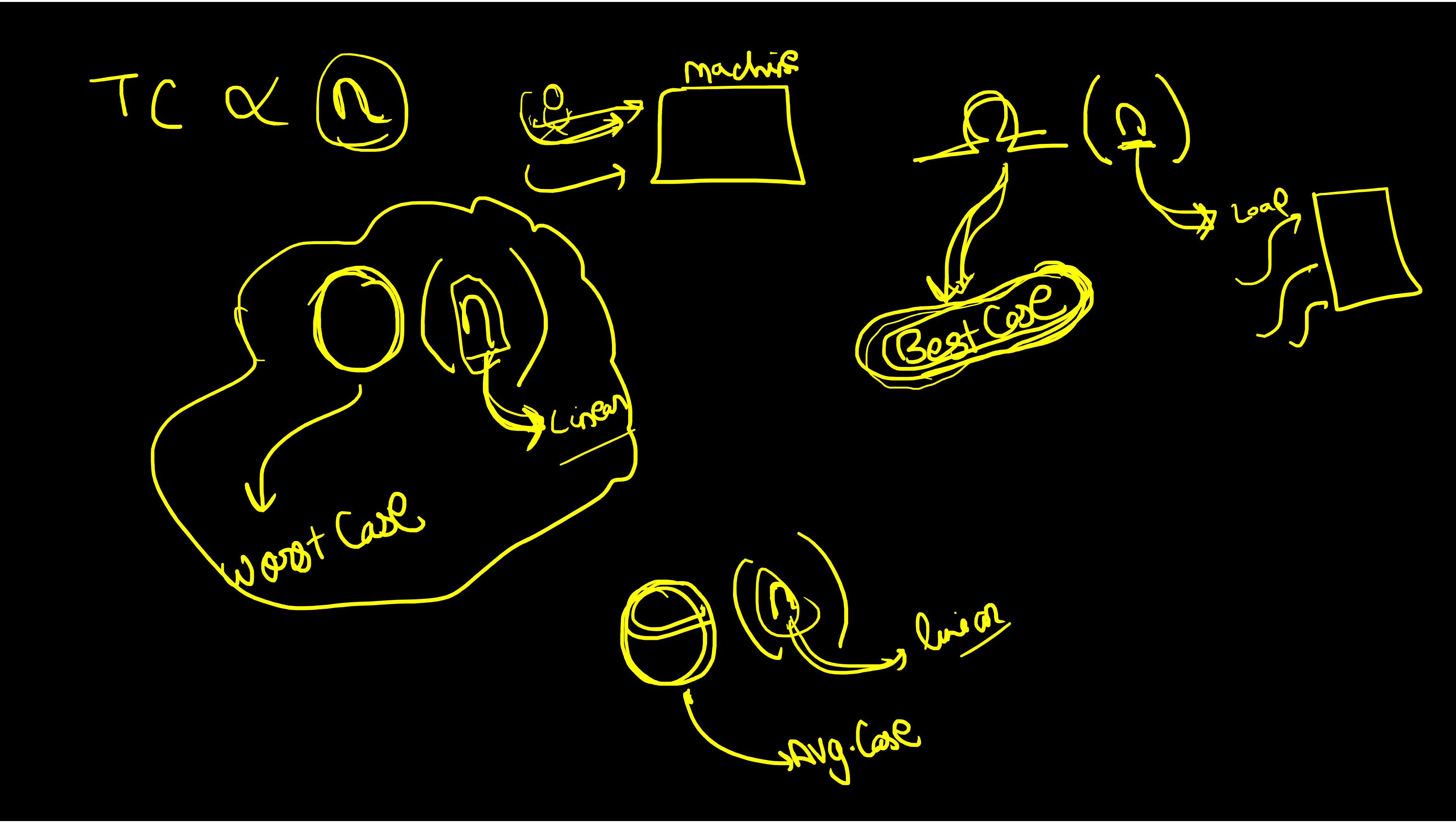


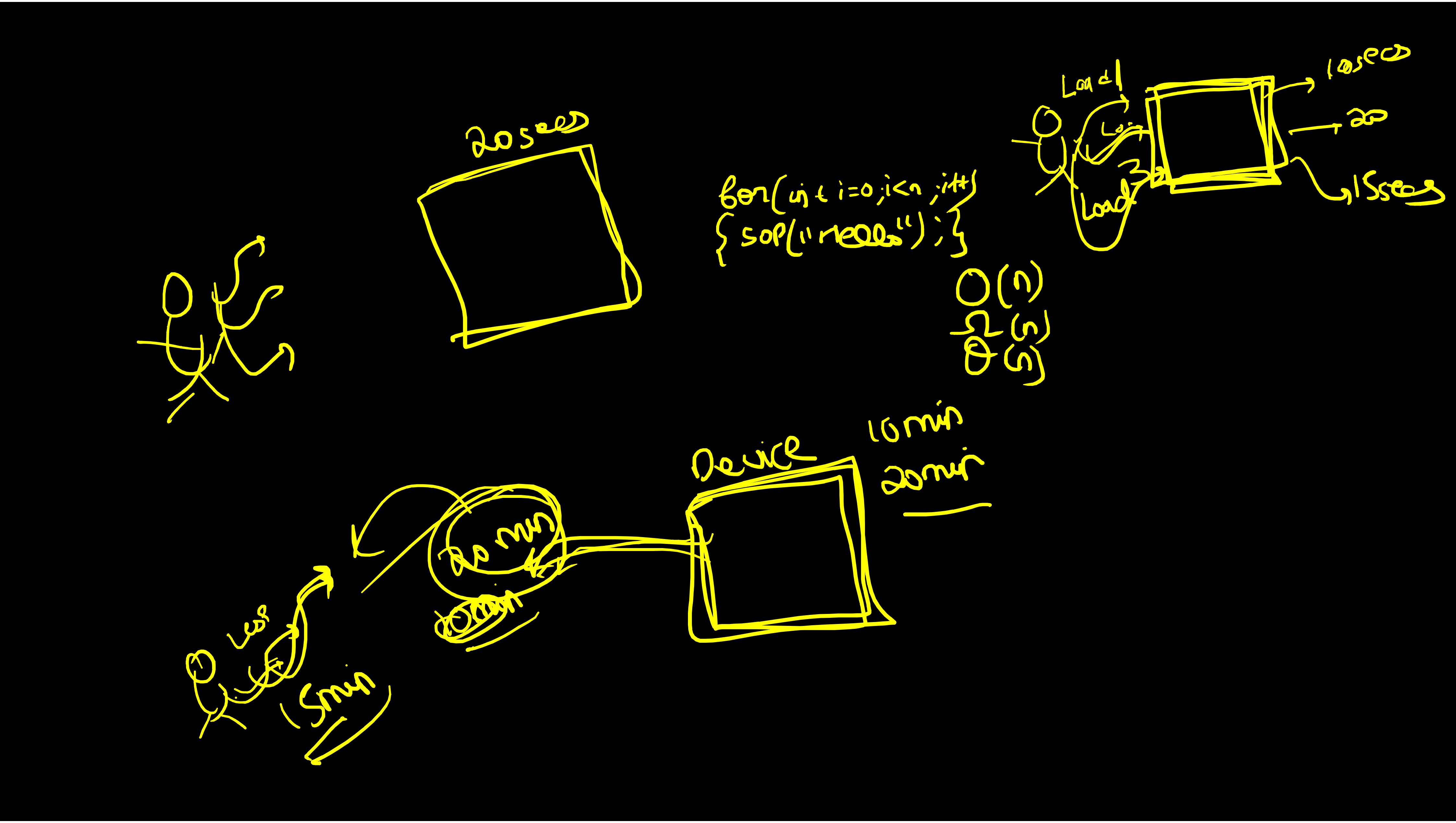
10000

$\sim n$



$S \propto n$


TC ∝ (n)
machine
Loop
Best Case
O (n)
Linear
Worst Case
O (n)
Linear
Avg. Case



20secs
for(int i=0;i<n;i++)
{sop("Hello");}
O(n)
Ω(n)
Θ(n)
Load
10secs
20
15secs
Load
Device
10min
20min
20 min
Req
15min

find target

target = 8

nums

7	5	6	7	1	2	3	4

target

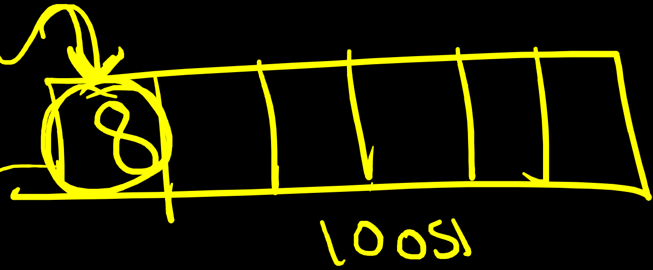
$O(n)$

target = 8

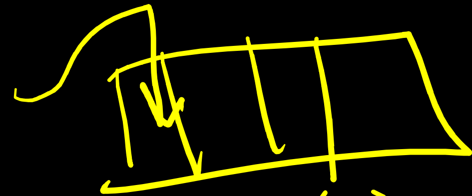
7	5	6	8	5	7

Linear Search → $O(n)$

target=8



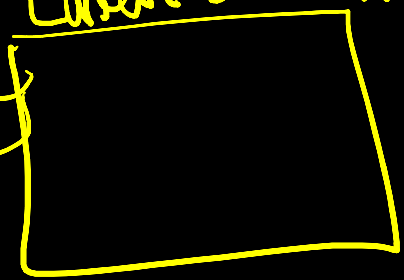
Best Case → $\Omega(1)$ → constant



$\Omega(1)$

n inspections

Linear Search



$O(n)$

Best case $O(1)$

Linear search

$O(n) \rightarrow$ W.C.

$O(1) \rightarrow$ B.C

$\Omega(1)$

Θ

$1+2+3+4+5\ldots+n$

$= \dfrac{\frac{n(n+1)}{2}}{n}$

$\Theta\left(\frac{n+1}{2}\right)$

$\left(\frac{n+1}{2}\right)$

$\Theta($

Scenario 1 Scenario 2 Se... Scenario n

1 2 3 n

$$1+2+3+4+5 \ldots +n$$

$$\frac{n(n+1)}{2}$$

n

$$\Theta\left(\frac{n+1}{2}\right)$$

$$\frac{n(n+1)}{2n}$$

$O($

Ω

Θ

$n \uparrow$

$10^{10} >> 10^5$

n

$O(n^2+3n+5)$ $\quad (10^5)^2 + 10^5$

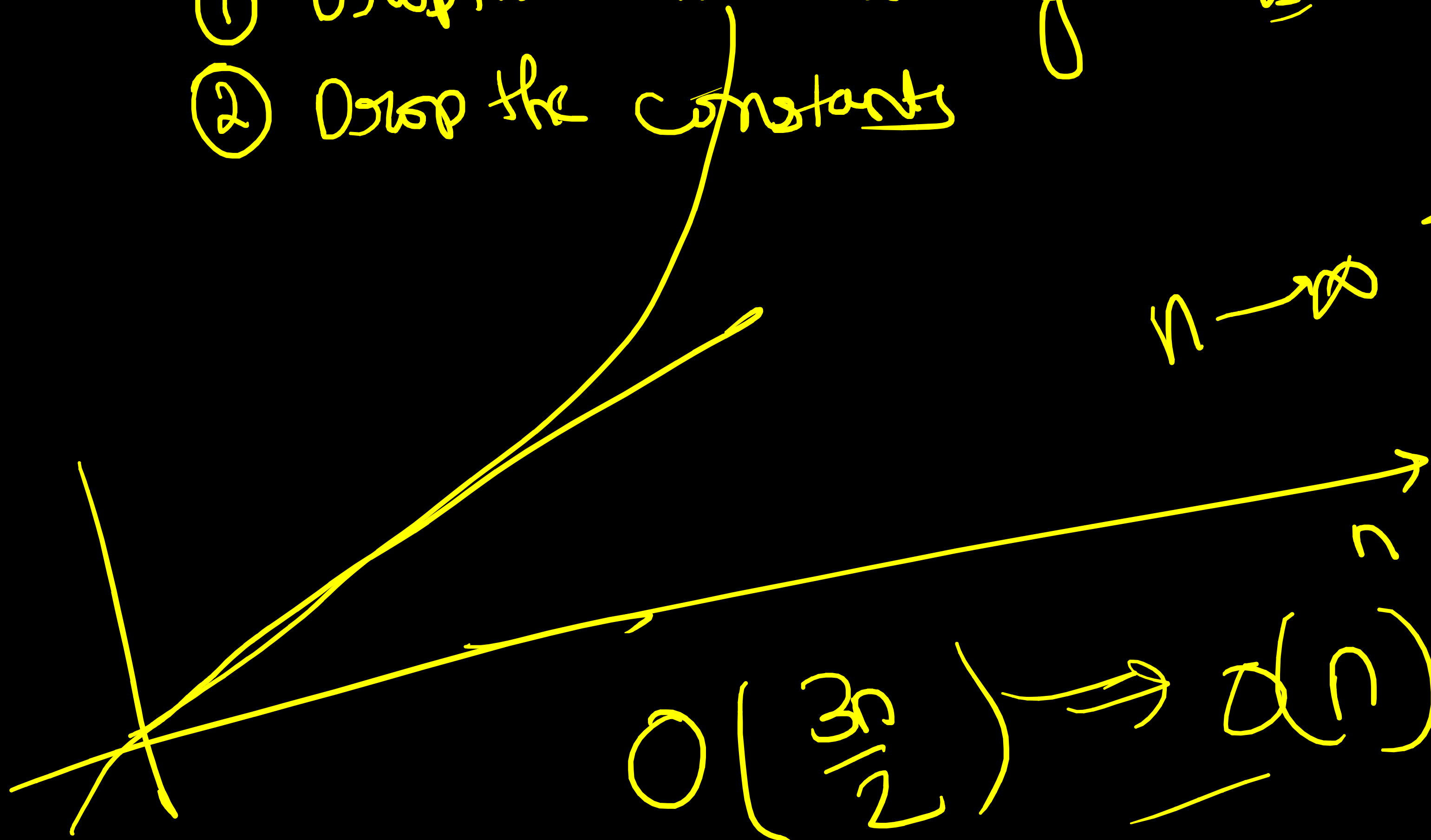
$O(n^2+\cancel{n})$

$O(n^2)$

$n^2 >> n$

1. Drop the non dominating terms
2. Drop the constants

$n \to \infty \quad \frac{10^{100}}{n^2} \qquad 10^5$



$O\left(\frac{3n}{2}\right) \to O(n)$

10000

10005

100

103

$O(\cancel{4}n^2 + \cancel{2}n + \cancel{5}) \rightarrow O(\frac{5}{2}n)$

$O(\frac{2}{3}n)$

$O(n^2)$

$O(n)$

$$O(n^2)$$

$O(n^3 + n^2)$

$O(n^3)$

$O(n^3 + n^2)$