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

i/p

main

fName()||

fName

a, c

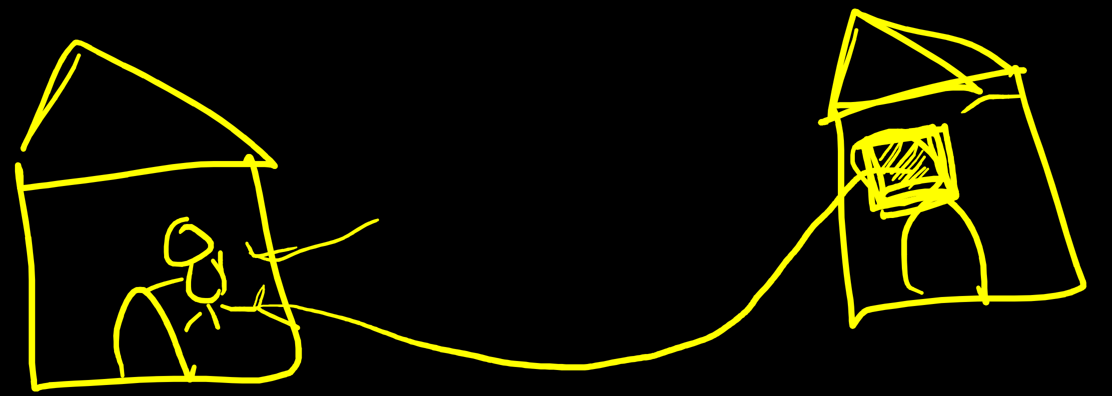
return a;

Hello

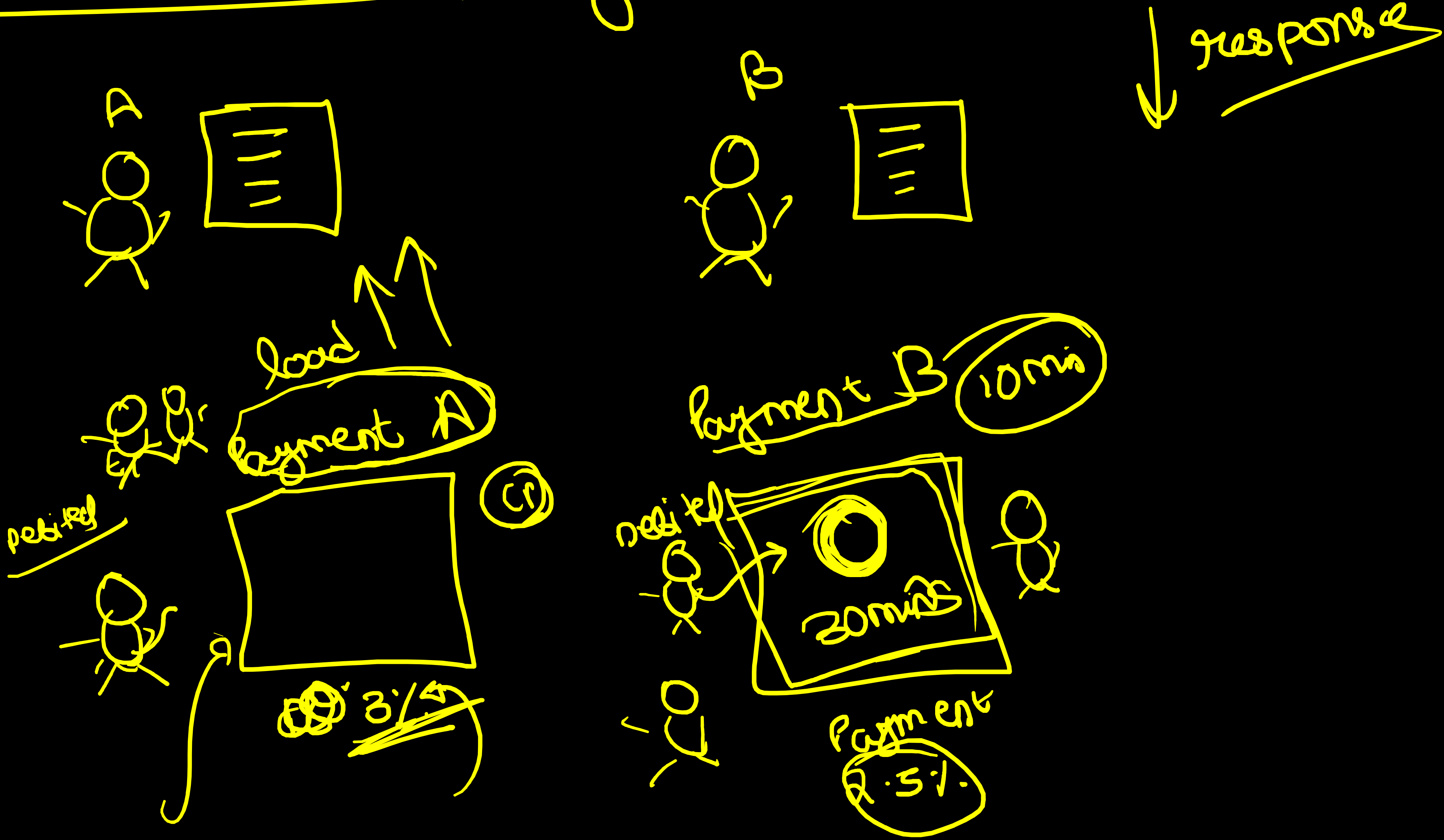
5

'A'

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



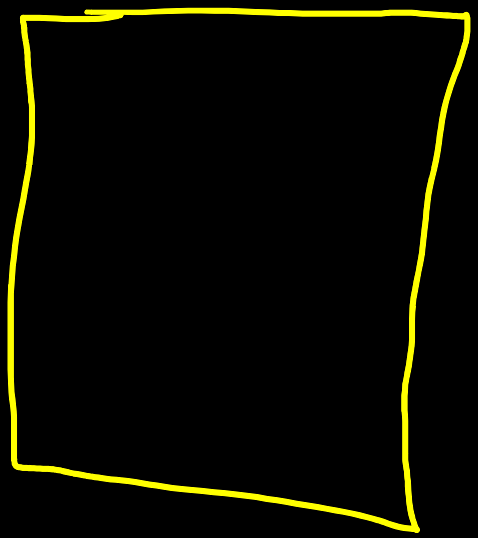
Time and Space Complexity



Space

program

RAM

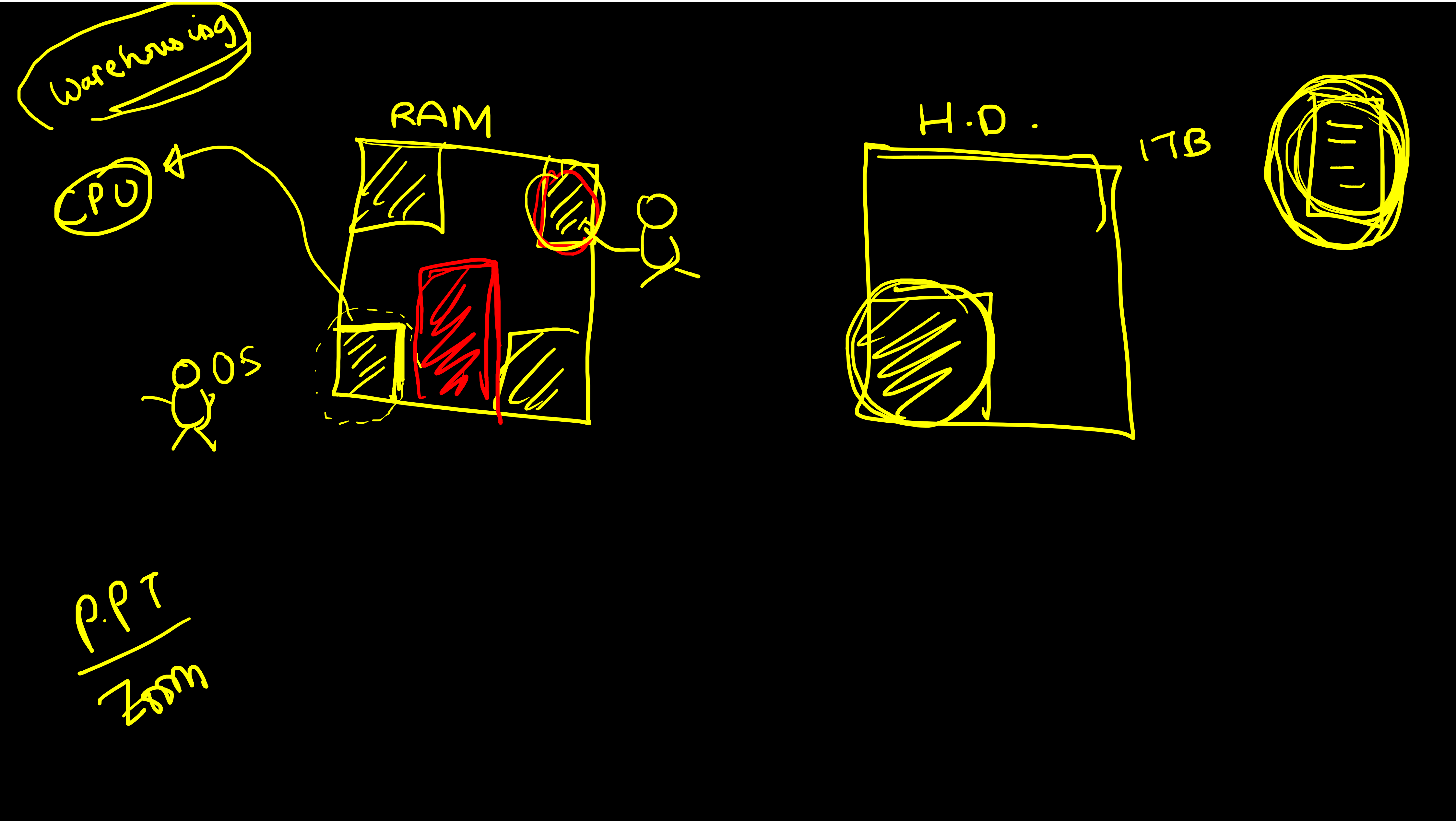


Application

Application

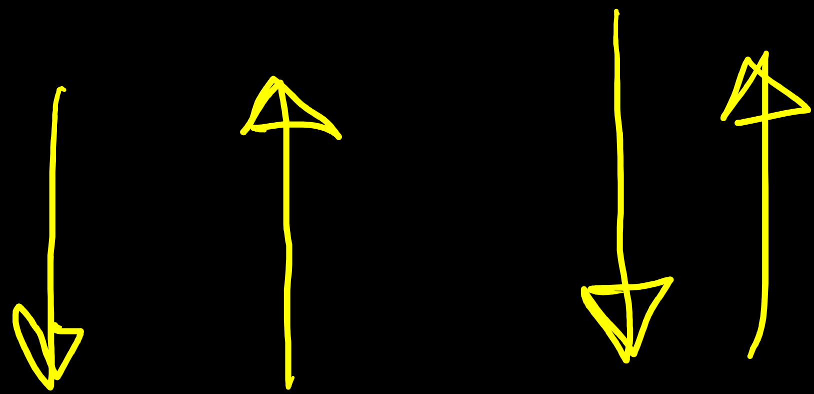
5MB

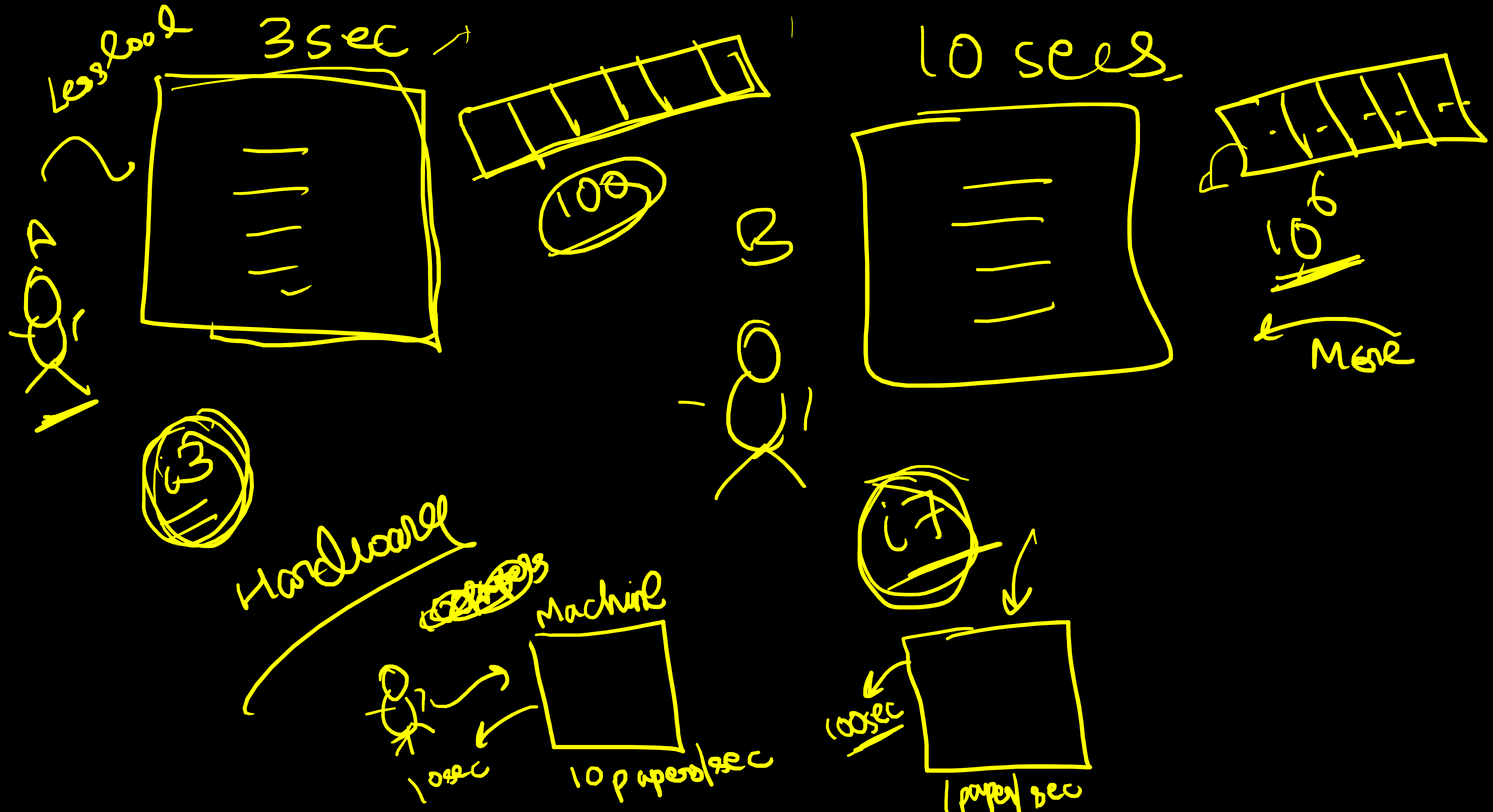
1GB


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

Time & Space

Time/Space

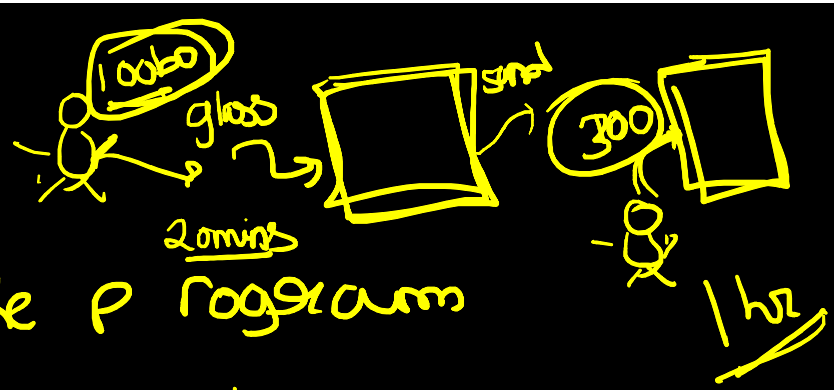




Less load
3 sec
100
A
10 secs
B
100
More
i3
i7
Hardware
Machine
10 sec
10 papers/sec
100 sec
1 paper/sec

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

$T \propto n$

$100 \rightarrow 500\text{sec}$

$1000 \rightarrow 5000\text{sec}$

$T \propto n$

n

10^5

$n^2 > n$

Machine B

n^2

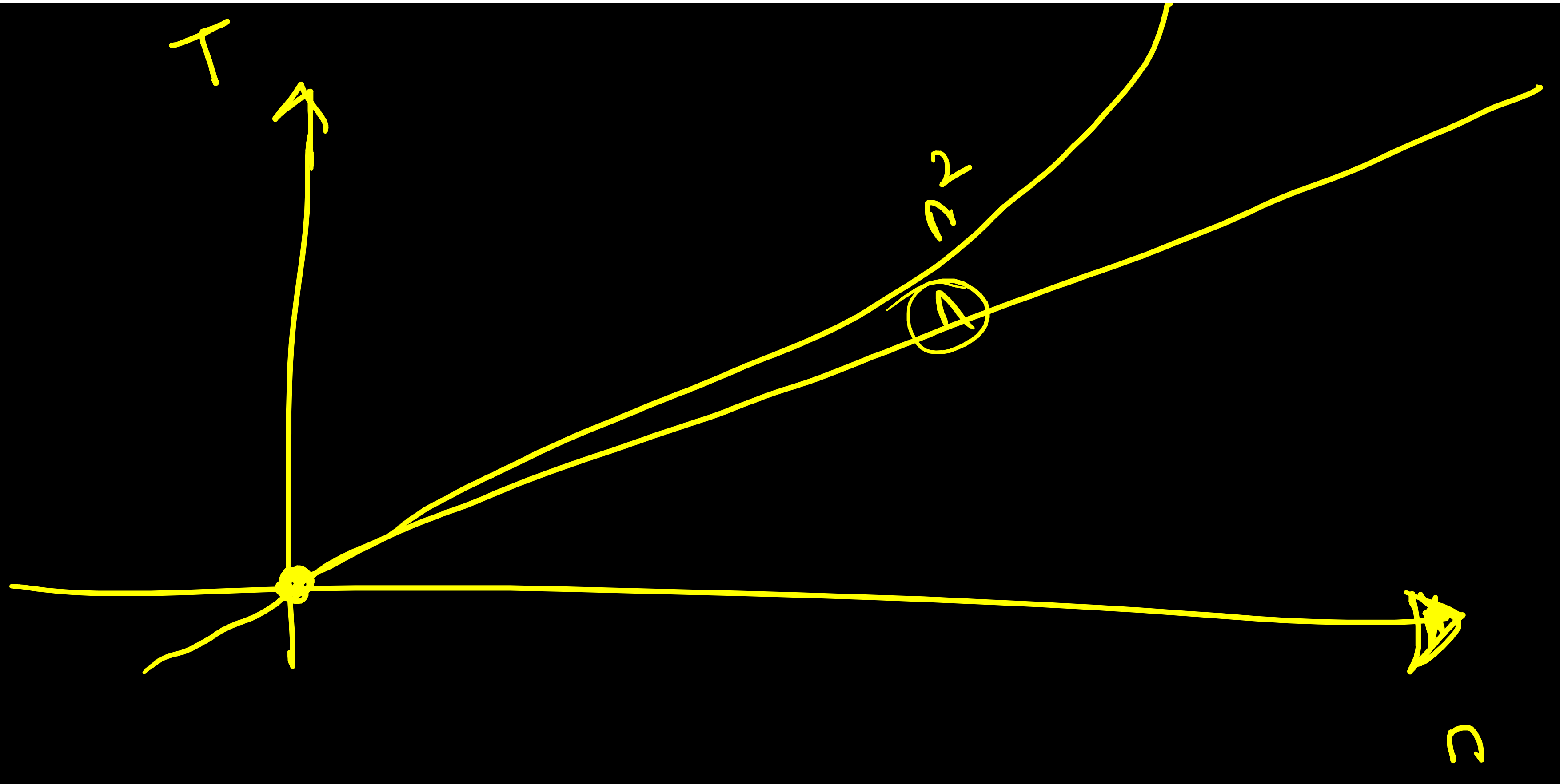
$5 \rightarrow 25\text{ secs}$

$10 \rightarrow 100\text{sec}$

$15 \rightarrow 225\text{s}$

$T \propto n^2$

10^{10}


T
2
n

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

$n = 10^5$

100 times

10^5

n

0

$T \propto n$

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

0 times

$n \uparrow$

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"); }
}
```

$n=3 \rightarrow 9$

$n=4 \rightarrow 16$

$T \propto n^2$

$i=1 \rightarrow n$ times

$i=2 \rightarrow n$ times

$i=3 \rightarrow n$ times

$i=4 \rightarrow n$ times

no

$\vdots$

$i=n \rightarrow n$ times

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

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

loop

if else

$i=0 \rightarrow$ n times

$i=1 \rightarrow$ n times

$i=2 \rightarrow$ n times

$\vdots$

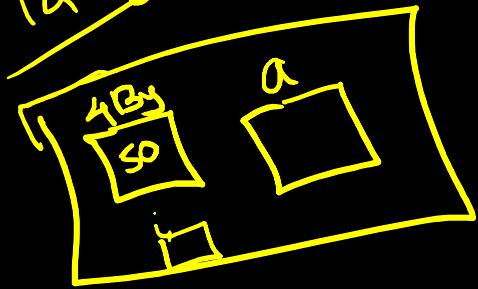
$i=m-1 \rightarrow$ n times

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

$m*n$

Space $\to f(n)$

12 Bytes



50

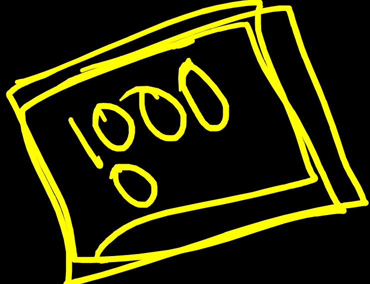
```
int n;
int a = 0;
for(int i=0; i<n; i++)
{a=a+1;}
```

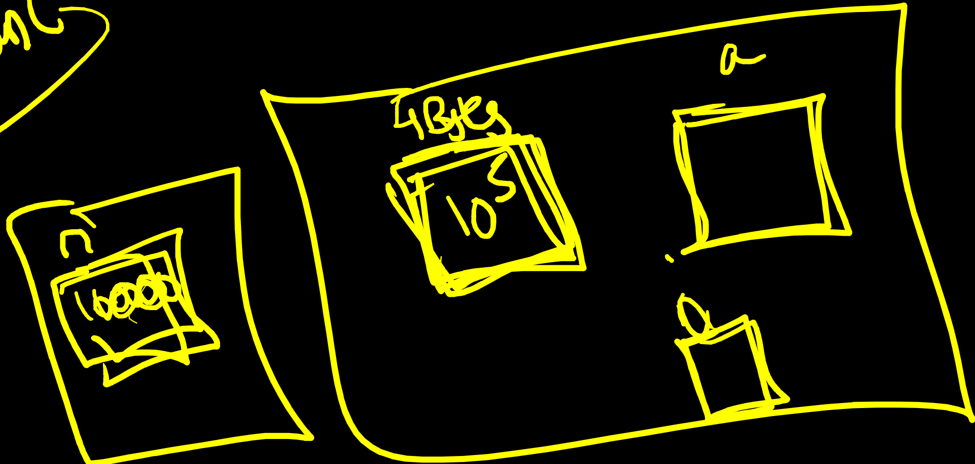
$2^{31}-1$

$n=10000$

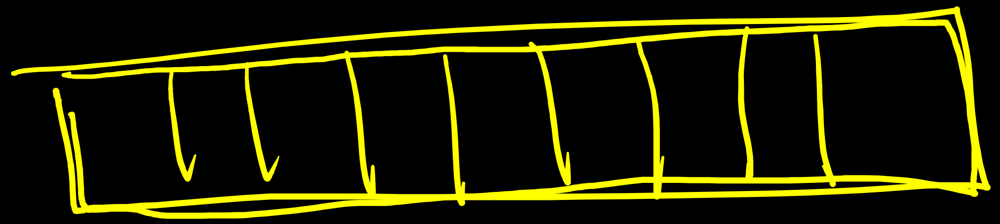
$n=100$

Space is Constant





$\sim n$



$S \propto n$

$TC \propto (n)$

machine

$\Omega(n)$ → Best Case

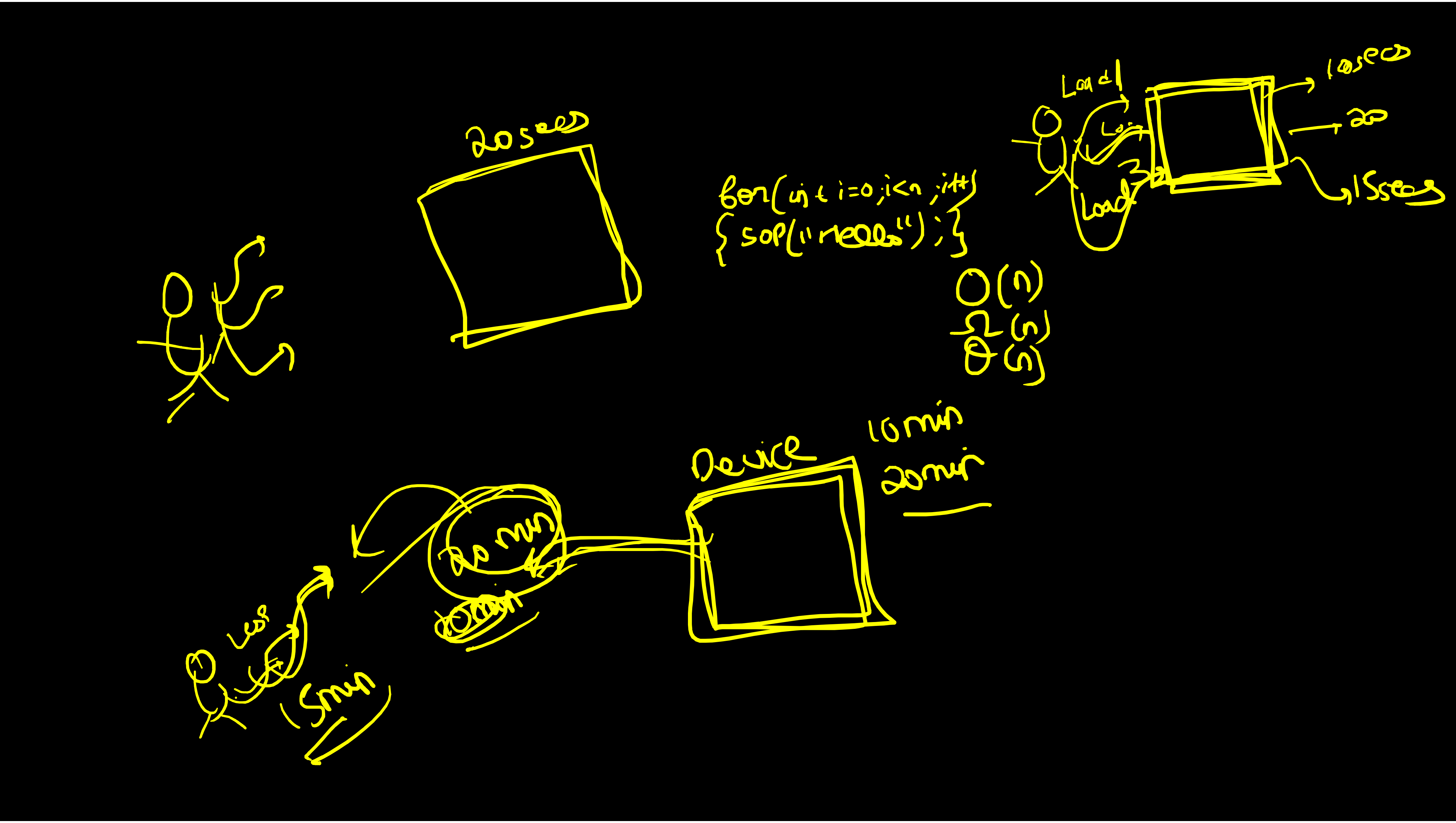
Load

$O(n)$ → Linear

Worst Case

$\Theta(n)$ → linear

Avg. Case


20secs
for(int i=0;i<n;i++)
{sop("hello");}
O(n)
Ω(n)
θ(n)
Load
Load
10secs
20
15secs
Device
10min
20min
20min
Less
5min

find target

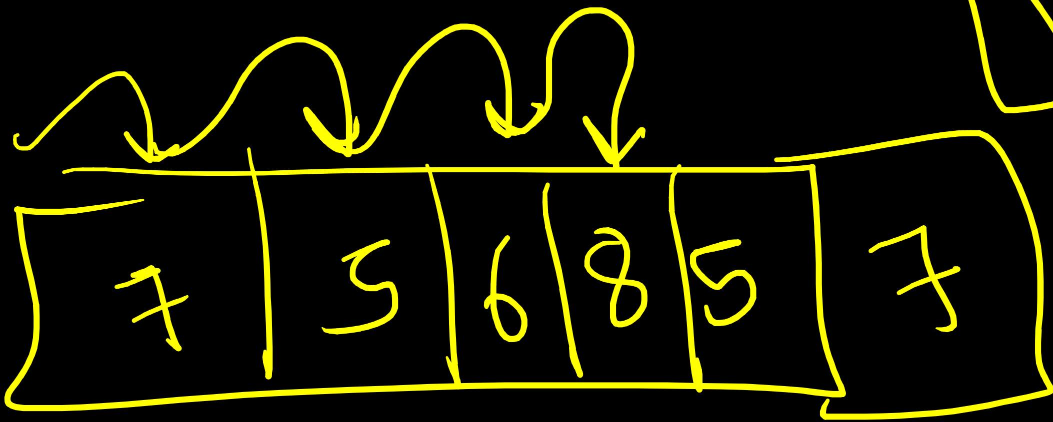
target = 8

n

target

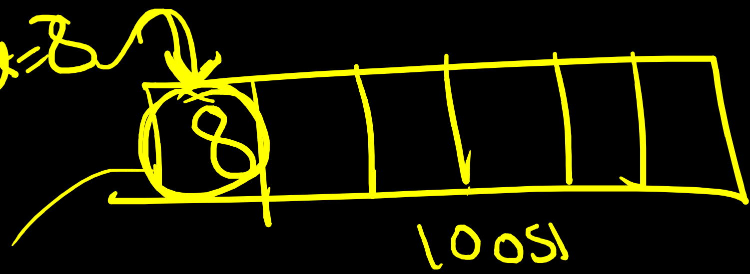
$O(n)$

target = 8

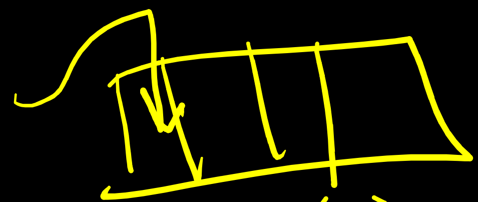


Linear Search → $O(n)$

target = 8



Best Case → $\Omega(1)$ → constant



$\Omega(1)$

n inspections

$\Omega(1)$

Linear Search

$O(n)$

Best case $O(1)$

$\leq n$

Linear search

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

$\Omega(1)$ $O(1) \rightarrow$ B C

O

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

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

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

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

$\Theta($

Scenario 1 Scenario 2 Sc Scenario n

```
for(int i=0;i<n;i++)
{ if (arr[i]==target)
  {return i;}
}
return -1;
```

1 2 3 n

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

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

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

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

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

$O($

Ω

Θ

$n \uparrow$ $n^2 >> n$

n

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

n

$O(n^2 + 3n + 5)$

$(10^5)^2 + 10^5$

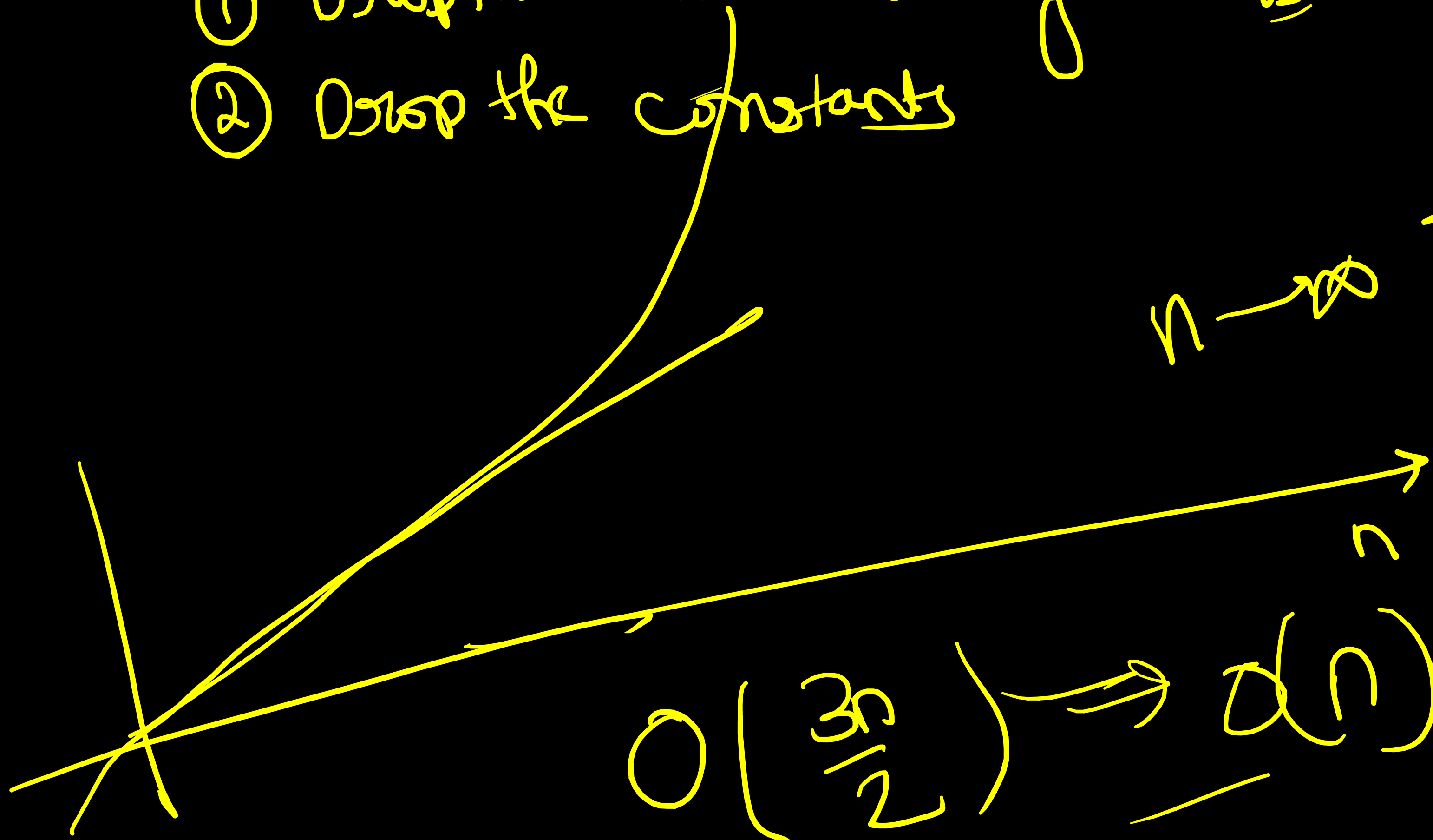
$n^2 >> n$

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

$O(n^2)$

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

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



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

10000
10005

100
103

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

$O(\frac{5}{2}n)$

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

$O(n)$

$O(n^2)$

$O(n^2)$

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

$O(n^3)$

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

for (i = 0 → m) → m times

{ SOP(i) } ✓ O(m)

m+n

O(m+n)

for (i = 0 → n) → n times

{ SOP(i } ✓

for (i 0 → n)
{ for(j 0 → n)
{ }
}
for(j = 0 → n)
{ }

$O(n^2 + n)$

$\Downarrow$

$O(n^2)$

$O(1)$

$O(n^2)$

for (i=0; i<n ; i++)
{ for (j=0; j<i ; j++) $\rightarrow$ i times
{ ( ) } $\rightarrow$
}

$\overbrace{0 \quad 1 \quad 2 \quad \ldots \quad i-1}^{i \text{ diff}}$ $j=0$

i=0 $\rightarrow$ 0 times
i=1 $\rightarrow$ 1 time
i=2 $\rightarrow$ 2 times
⋮
i=n-1 $\rightarrow$ (n-1) times

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

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

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

$\{ \underbrace{1+2+3+ \cdots (n-1)}_{} \}$

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

$O(n^2)$

$1 + 2 + 3 + 4 + \dots (n-1)$

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

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

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

$O(n^2)$

$$O\left(\frac{n^2}{\cancel{2}} - \cancel{\frac{n}{2}}\right)$$

for(int i=n; i>=1; i/=2)

{ }

i=8

8

4

2

1

9

4

2

1

n

$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

$\frac{n}{16}$

1

for(i=n; i>=1; i/=3)

$O(\log_3 n)$

n

$\frac{n}{2}$ ✓

$\frac{n}{4}$ ✓

$\vdots$

1 ✓

n

$\frac{n}{2^1}$

$\frac{n}{2^2}$

$\frac{n}{2^3}$

$\vdots$

1

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

$$\Rightarrow 2^k = n$$

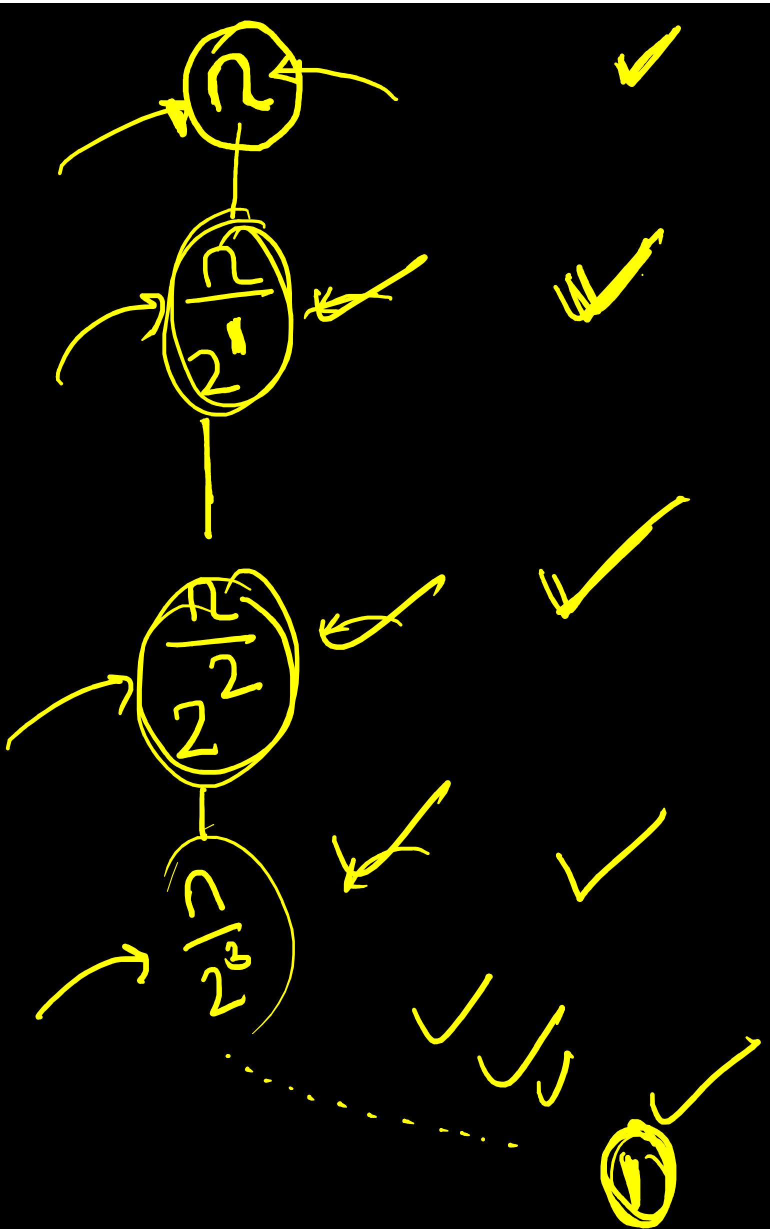
$$\Rightarrow k = \log_2 n$$

$$O(k+1)$$

$\underline{(k+1)}$

$$= O(\log n + 1)$$

$$= O(\log n$$



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

$$\Rightarrow 2^k = n$$

$$\Rightarrow \boxed{k = \log_2^n}$$

$(K+1)$

i=0 ; i<1

i=0 { = }

$$O(\log_2 n + 1)$$

for(i=n; i>=1; i/=m)

$O(\log_m n)$

i*=2

log

```
i = n
while(i>=1)
{ i/=2;
  a=a+1;
}
```

$O(\log_2 n)$

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

$j = 1$

$n+1$

$j = 1, 2, 3, 4 \ldots n$

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

$i=1 \rightarrow n$ times

$i=2 \rightarrow \frac{n}{2}$ times

$i=3 \rightarrow \frac{n}{3}$ times

$i=4 \rightarrow \frac{n}{4}$

$\vdots$

$i=n \rightarrow \frac{n}{n}$

$1 \quad 3 \quad 5 \quad 7 \quad 9$

$1 \quad 4 \quad 7$

$$O\left(n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \frac{n}{5} + \cdots\cdots 1\right)$$

$$O\left(\frac{n^2}{\alpha}\right)$$

$$\boxed{O(n \log n)}$$

$$n\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots\cdots \frac{1}{n}\right)$$

α

$$O(n \log n)$$

$$\int_1^n \frac{1}{x}\,dx = \log x\Big|_1^n = \log n - \underline{\log 1}$$

$$= \log_e n$$

$\log n$

$\ln x$

e

$$\frac{1}{x} \quad \frac{1}{1} + \frac{1}{1.00001} + \frac{1}{1.001} + \frac{1}{1.1} + \frac{1}{1.5} + \frac{1}{2} + \frac{1}{2.00} + \frac{1}{3} + \frac{1}{4}$$

$$+ \ldots \frac{1}{n}$$

$$\underbrace{1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4} \cdots\cdots} < \log_2 n \qquad \underline{O(\ )}$$

$$\underbrace{n\left(1+\frac{1}{2}+\frac{1}{3}+\cdots\cdots\right)}$$

$$O(n \log n)$$

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

$$\log n < n$$

$$\sqrt{n} > \log n$$

$$\log \sqrt{n}$$

$$\Rightarrow \frac{1}{2} \log n$$

$$\log(\log n)$$

$$\log(\underbrace{\log n})$$

$$g(n) = \begin{cases} n & n < 1000 \\ \underline{n^2} & n >= 1000 \end{cases}$$

$$\underline{O(n^2)}$$

$$g_1(n) = \begin{cases} n & n <= 100 \\ \underline{n^4} & n > 100 \end{cases}$$

$$\underline{g_1 + g_2}$$

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

$$g_2(n) = \begin{cases} n^3 & n <= 100 \\ \boxed{n^2 \quad n > 100} \end{cases}$$

$$\underline{O(n^4)}$$

$O(1) \longrightarrow$ T.C

$O(1)$

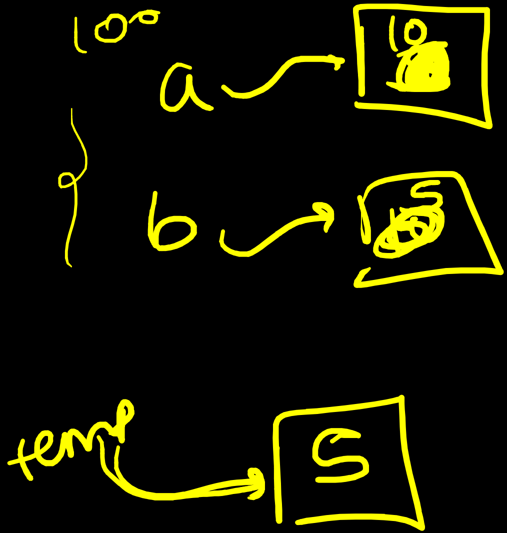
max a, b

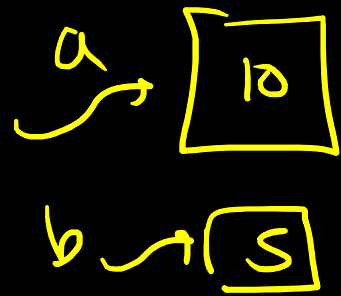
```
if (a>b)
{ max=a; }
else
{max=b;}
```

Swap 2 variables

Swap

```
int a = 5;
int b = 10;
int temp = a;
a = b;
b = temp;
```





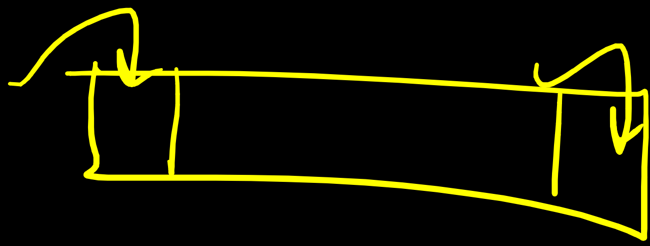
O(1)

O(1)

5

a = ;
b = 5;
a = b;

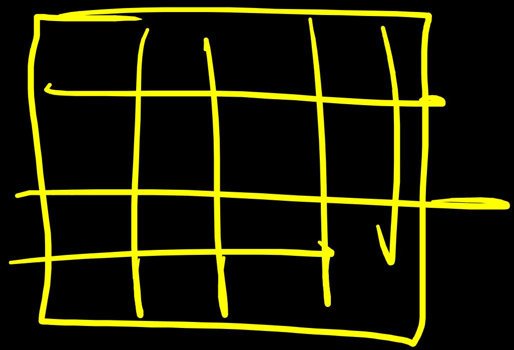
sum of first and last element



$O(1)$

$sum = arr[0] + arr[n-1]$

m rows,
n colume



$O(m*n)$

for(i=1; i<=n ; i*=2)

{ }

$O(\log_2 n)$

$1 \longrightarrow 2^1 \longrightarrow 2^2\ 4 \longrightarrow 8 \longrightarrow 2^k$

min

$2^k = n$

$\Rightarrow k = \log n$

```
for ( i=1 ; i<=n ; i++)      <- n
{ for( j=1 ; j<=n ; j*=2)    <- log n
    {
    }
}
```

$O(n \log n)$

for(i=0; i<n; i++) n

{ for(j=0; j<n; j++)

{ sopl(Hello)

break;

}

}

$i=0 \rightarrow 1$ times

$i=1 \rightarrow 1$ tim

$i=2 \rightarrow 1$

$\vdots$

$i=n-1 \rightarrow 1$ times

$$\underbrace{1 + 1 + 1 + 1}_{n \text{ times}}$$

$O(n)$

```
for( i= 0; i<n ; i++)
{ for( j=0 ;j<n  ;j++)
  { if( i<n-1)
    {break;}
  }
}
```

n-1

$i=0 \longrightarrow 1$

$i=1 \longrightarrow 1$

$\vdots$

$i=n-2 \longrightarrow 1$

$i=n-1 \longrightarrow$ n times $\to O(n)$

$$\underbrace{1+1+\ldots1+1}_{(n-1)} + n$$

$$(n-1)+n \quad \boxed{O(2n-1)}$$

$O(n)$

$O(n^2)$

$O(n^3)$

$O(\log n)$

$O(n \log n)$

$O(1)$

$$O(1) < O(\log n) < O(\sqrt{n}) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n) < O(3^n)$$

$n \log n > n$

$O(n^2)$ $O(n)$

```
int a = 0;
int i = N;
while (i>0)
{ a+=i;
  i/=2;
}
int[] arr = new int[a];
```

$$T.C \rightarrow \boxed{O(\log_2 n)}$$

$O(a)$

N — $\frac{N}{2}$ — $\frac{N}{4}$ — ...

$$a = N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \ldots 1$$

1

$$a = N + \frac{N}{2} + \frac{N}{4} + \ldots$$

$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}$

$$n+\frac{n}{2}+\frac{n}{4}+\frac{n}{8}+\frac{n}{16}+\cdots\cdots 1$$

$n \uparrow \uparrow$

$$\Rightarrow n\left(1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\cdots\cdots\frac{1}{n}\right) \rightsquigarrow \text{G.P}$$

$\rightarrow a=1$

$\rightarrow r=1/2$

1

$$\frac{a}{1-r}$$

$$\boxed{a=2n}$$

$a=1$

$r=\frac{1}{2}$

$$=\frac{1}{1-\frac{1}{2}}=\underline{2}$$

$a \quad r$

$$\frac{a}{1-r}$$

$$\underline{O(2n) \rightsquigarrow O(n)}$$

n

If this no. is a perfect square

n

1, 4, 9, 16, 25, 36, 49

$4*4=16$

$i*i==n$

Perfect square $\rightarrow O(\sqrt{n})$

Non-perfect $\rightarrow O(n)$

```
for( i=1; i<n; i++)
{ if( i*i ==n)
   { return true;}
}
return false;
```

$\boxed{O(n)}$

$i = \sqrt{n}$

7 . . .

$i = \sqrt{n} * \sqrt{n} = n$

n

perfect

$i = \sqrt{n}$

$i * i$

n

non.perf

$i = \sqrt{n}$

50

1, 2, 3, 4, . . . 7, 8

$(\sqrt{n} + 1)(\sqrt{n} + 1) > n$

Math.sqrt

```
{
for(i=1; i<=√n ; i++)
{ if(i*i==n)
    {return true;}
}
return false;
}
```

$\sqrt{n}$

$O(\sqrt{n})$

i=1 . . .

$\sqrt{n} * \sqrt{n} = n$

$(\sqrt{n}+1)(\sqrt{n}+1) > n$

```
for(i=1; i<n; i++)
{ if (i*i == n)
    {return true;}
  if (i*i > n)
    {return false;}
}
```

$\sqrt{n}$

$O(\sqrt{n})$

$\sqrt{n} + 1$