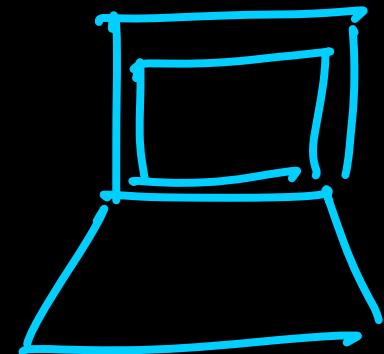
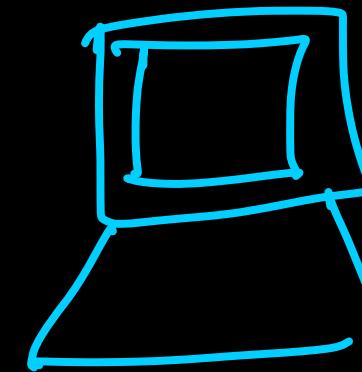


# Time Complexity

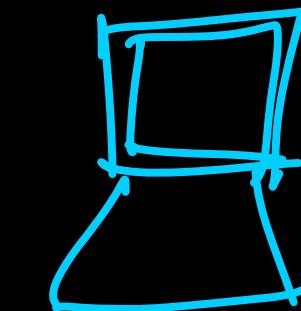
# Time Complexity



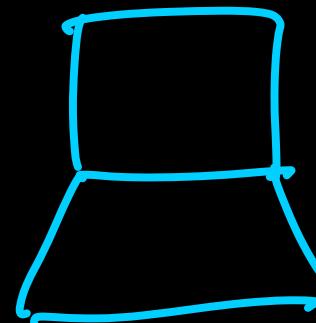
i9  
32GB



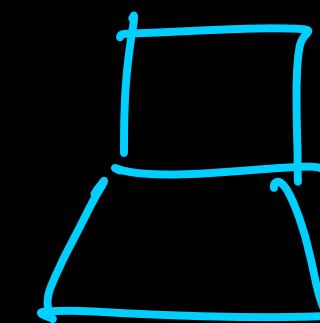
i3  
8GB



i5  
16GB

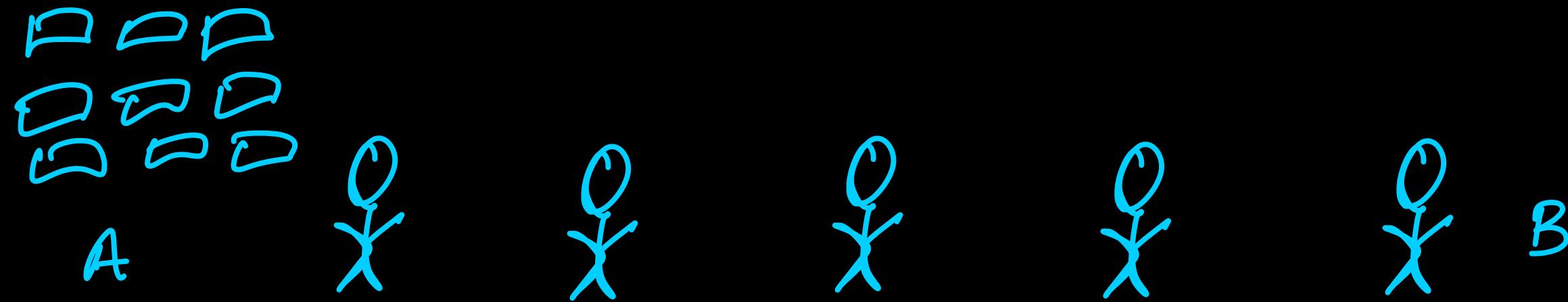


i3



i3

# Time Complexity



Fact code  $\neq$  less no. of lines

**Ques :** Given an array of size  $n+1$  consisting of integers from 1 to  $n$ . One of the elements is duplicate in the array. Find that duplicate element.

arr = { 5, 1, 3, 4, 2, 4 }

No. of operations

# Approach 1 → brute force

arr = { 5, 1, 3, 4, 2, 4 }

No. of ops = 5 + 4 + 3 + 2 = 14

S.C. =  $O(n)$

A.S. =  $O(1)$

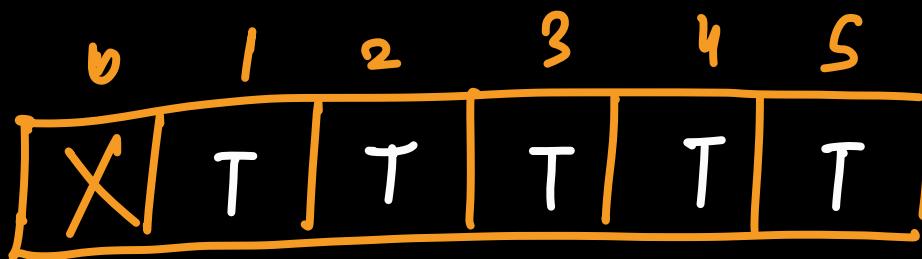
T.C. =  $O(n^2)$

```

for(int i=0; i<n; i++) {
    for(int j=i+1; j<n; j++) {
        if(arr[i] == arr[j]) {
            mil gaya
        }
    }
}
  
```

## Approach 2 → marking in another array

arr = { 5, 1, 3, 4, 2, 4 }

flag = 

```
boolean[] flag = new boolean[n+1];
for(int i=0; i<n; i++){
    int ele = arr[i];
    if(flag[ele]==true) imposter
    else flag[ele]=true
}
```

Total no. of ops = 6

A.S. =  $O(n)$

S.C. =  $O(n)$

T.C. =  $O(n)$

## Approach 3 'Most Optimised'

$arr = \{ 5, 1, 3, 4, 2, 4 \}$        $arr.length = 6 \rightarrow 1 \text{ to } S \text{ numbers}$

$$\text{Sum Array} = 5 + 1 + 3 + 4 + 2 + 4 = 19$$

$$\text{sum 1 to } S = \frac{S * 6}{2} = 15$$

$$A.S. = O(1)$$

$$T.C. = O(n)$$

$$19 - 15 = 4$$

$$\text{Total no. of ops} = 6$$

## Ques: Calculate the time complexity

```
for(int i = 1; i <= n; i++) {  
    sout("GFG");  
}
```

GFG will be printed 'n' times

This loop has 'n' iterations

Total ops = n

Time Complexity = O(n)

## Ques: Calculate the time complexity

```
for(int i = 1; i <= n; i+=2) {  
    sout("GFG");  
}
```

Kitni bar GFG print hoga  $\approx \frac{n}{2}$  times

Total no. of ops =  $\frac{n}{2}$

T.C. =  $O\left(\frac{n}{2}\right) = O(n)$

$$O(k \cdot n) = O(n)$$

## Ques: Calculate the time complexity

```
for(int i = 1; i <= n-7; i++) {  
    sout("GFG");  
}
```

Total no. of ops =  $n-7$

$$T.C. = O(n-7) = O(n)$$

---

$$O(3n+8) = O(n)$$

$$O(n \pm k) = O(n)$$

## Ques: Calculate the time complexity

```
for(int i = 1; i <= n; i++) {  
    sout("GFG");  
}
```

$$T \cdot n \cdot O \cdot = 2n$$

$$T \cdot C \cdot = O(2n) = O(n)$$

```
for(int i = 1; i <= n; i++) {  
    sout("GFG");  
}
```

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

```
    sout ("khushi");
```

```
    sout ("Vineet");
```

```
    sout ("Archit");
```

}

$T.C. = O(n)$

$T.n.O = 3n$

## Ques: Calculate the time complexity

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

```
    cout("GFG");
```

$$T \cdot n \cdot O = n + m$$

```
}
```

```
for(int i = 1; i <= m; i++) {
```

```
    cout("GFG");
```

$$T \cdot C = O(m + n)$$

```
}
```

```
for (int i=1; i<=200; i++) {
```

```
    cout ("Somil");  
}
```

$$T \cdot n \cdot o = 200$$

$$T \cdot C \cdot = O(200) = O(1)$$

$$T \cdot C \cdot = ?$$

Constant time

$O(n) \rightarrow$  Linear Time

H.W.

```
for(int i=-8 ; i<n+5 ; i+=3){  
    cout("your name");  
}
```

# Space Complexity & Auxiliary Space

↓  
Extra space taken to solve the problem

Whenever we use extra data structures

## Ques: Calculate the time complexity

```
for(int i = 1; i <= m; i++) {
```

$$T \cdot n \cdot O \cdot = m^* n$$

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

$$T.C. = O(m^* n)$$

```
        sout("GFG");
```

```
}
```

```
}
```

$i = 1 \rightarrow j = 1, 2, 3, 4, \dots, n$

$i = 2 \rightarrow j = 1, 2, 3, \dots, n$

!

:

:

$i = m \rightarrow j = 1, 2, 3, \dots, n$

## Ques: Calculate the time complexity

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

$$T.C. = O(n^2)$$

Quadratic Time

## NESTING of Loops

```
for(i → 1 to n){  
    for(j → 1 to m){  
        for(k → 1 to p){  
            for(l → 1 to q){  
                cout(b, F, b)  
            }  
        }  
    }  
}
```

$$T.C. = O(n * m * p * q)$$

## Ques: Calculate the time complexity

```

for(int i = 1; i <= n; i++) {
    for(int j = 1; j <= i; j++) {
        sout("GFG");
    }
}

```

$i=1 \rightarrow j=1$   
 $i=2 \rightarrow j=1, 2$   
 $i=3 \rightarrow j=1, 2, 3$   
 $\vdots$   
 $i=n \rightarrow j=1, 2, 3, \dots, n$

$$T \cdot n \cdot O \cdot = 1 + 2 + 3 + 4 + \dots + n = \frac{n(n+1)}{2} \text{ or } \frac{n^2+n}{2}$$

$$T \cdot C \cdot = O\left(\frac{n^2+n}{2}\right) \approx O(n^2) \approx O(n^2)$$

~~Ques~~  
~~HW~~



# HW: Calculate the time complexity

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

$i=1 \rightarrow j = 2, 3, 4, \dots n$   $n-1$

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

$i=2 \rightarrow j = 3, 4, 5, \dots n$   $n-2$

```
        sout("GFG");
```

$i=3 \rightarrow j = 4, 5, 6, \dots n$   $n-3$

```
}
```

$\vdots$

```
}
```

$i=n-1 \rightarrow j = n$

$1$

$i=n \rightarrow \infty$

$$T \cdot n \cdot O = 1 + 2 + 3 + \dots + n-1 = \frac{(n-1) \cdot n}{2} = \frac{n^2 - n}{2}$$

$$T \cdot C \cdot = O(n^2)$$

# \*Ques: Calculate the time complexity

```
for(int i = 1; i <= n; i*=2) {  
    sout("GFG");  
}
```

$x = ?$

$$a \cdot r^{x-1} = n \quad (x^{\text{th}} \text{ term})$$

$$1 \cdot 2^{x-1} = n$$

T.C. = 'i' ki value jithi baar change hogi

$$x-1 = \log_2 n$$

$$x = \log_2 n + 1$$

$i = 1, 2, 4, 8, 16, 32, \dots, n$



total 'x' terms

$$T.C. = O(x) = O(\log_2 n) = O(\log n)$$

$$\log_2 n = \frac{\log n}{\log 2} \Rightarrow O(\log_2 n) = O\left(\frac{\log n}{\log 2}\right) \underset{\sim}{=} O(\log n)$$

---

M.W. 1) `for (int i=n ; i>=1 ; i/=2) {`

    | `sout("Vinay");`  
    3

2) `for (int i=1 ; i<=n ; i*=3) {`

    | `sout ("Ananya");`  
    3

# HW: Calculate the time complexity

```
for(int i = 1; i <= n; i+=i) {  
    sout("GFG");  
}
```

## Ques: Calculate the time complexity

```
for(int i = 1; i <= n; i*=2) {           log n
    for(int j = 1; j<= n; j++) {           n
        sout("GFG");
    }
}
```

$$T.C. = O(n \log n)$$

Fast kya hai ?

$1 < \log n < n < n \cdot \log n < n^2 < n^3 <<< 2^n$

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

↓

Fastest

↓

Slowest

TLE

Time limit exceeded

# \* Ques: Calculate the time complexity

```

for(int i = 1; i <= n; i*=2) {
    for(int j = 1; j<= i; j++) {
        sout("GFG");
    }
}

```

$i=1 \rightarrow j=1$

$i=2 \rightarrow j=1, 2$

$i=4 \rightarrow j=1, 2, 3, 4$

$i=8 \rightarrow j=1, 2, 3, 4, 5, 6, 7, 8$

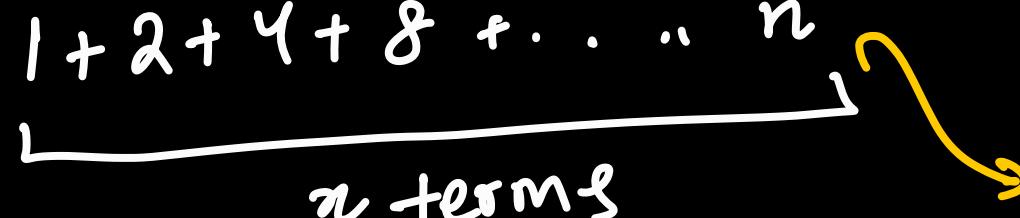
.

.

.

$i=n \rightarrow j= 1, 2, 3, 4, \dots, n$

$$T \cdot n \cdot O = 1 + 2 + 4 + 8 + \dots + n$$



$$\frac{a(r^x - 1)}{r-1} = \frac{1(2^n - 1)}{2-1} = 2^n - 1 = 2n - 1$$

$$2^x = 2n \leftarrow 2^{x-1} = n \Rightarrow x-1 = \log_2 n$$

T.C. =  $O(n)$

## \* Ques: Calculate the time complexity

```
for(int i = 1; i <= n; i*=2) {  
    for(int j = 1; j<= i; j++) {  
        sout("GFG");  
    }  
}
```

$$T(n) = 1 + 2 + 4 + 8 + \dots + n$$

$i=1 \rightarrow j=1$   
 $i=2 \rightarrow j=1, 2$   
 $i=4 \rightarrow j=1, 2, 3, 4$   
 $i=8 \rightarrow j=1, 2, 3, 4, 5, 6, 7, 8$   
⋮  
 $i=n \rightarrow j=1, 2, 3, 4, \dots, n$

$$S = 1 + 2 + 4 + 8 + 16 + \dots + n$$

$$S+1 = \underbrace{1+1}_{2+2+4+8+16+\dots+n} + 2 + 4 + 8 + 16 + \dots + n$$

$$2+2+4+8+16+\dots+n$$

$$4+4+8+16+\dots+n$$

$$\underbrace{8+8+\dots+n}_{16}$$

$$S+1 = 2n$$

$$S = 2n - 1$$

## Ques: Calculate the time complexity

```
for(int i = 1; i*i <= n; i++) {  
    sout("GFG");  
}
```

$i = 1, 2, 3, 4, \dots, \sqrt{n}$


$$i^2 \leq n$$

$$i^2 \leq n$$

$$i \leq \sqrt{n}$$

$$T \cdot n \cdot \theta = \sqrt{n}$$

$$T \cdot C \cdot = O(\sqrt{n})$$

~~Ques~~: Calculate the time complexity

```
for(int i = 1; i*i <= n; i*=2) {  
    sout("GFG");  
}
```

## \*\*Ques: Calculate the time complexity

```
for(int i = 2; i <= n; i*i = i) {  
    sout("GFG");  
}
```

JSKO karke di knao..

This was my best class

THANKYOU  
Cuties