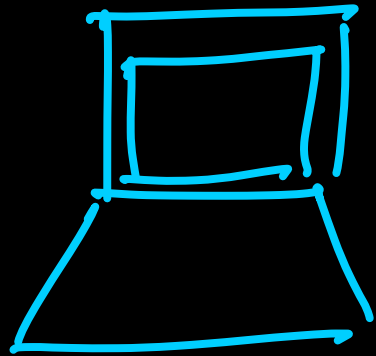


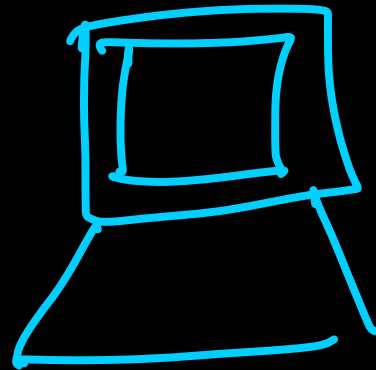


Time Complexity

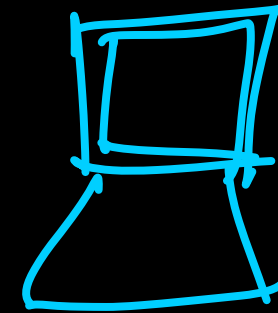
Time Complexity



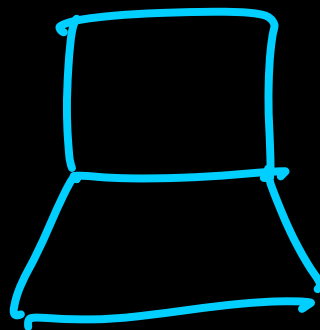
i9
32GB



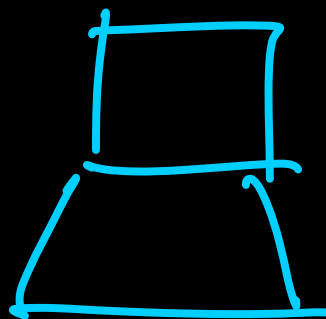
i3
8GB



i5
16GB

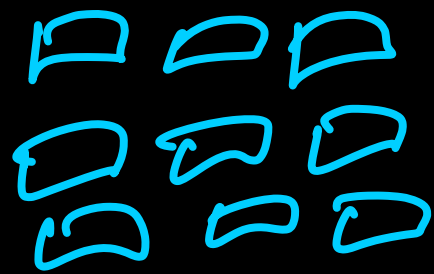


i3



i3

Time Complexity



A



B

Fast 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 \rightarrow 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
            | 3
        }
    }
}
```



Approach 2 → marking in another array

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

i

flag =

0	1	2	3	4	5
X	T	T	T	T	T

Total no. of ops = 6

A.S. = $O(n)$

S.C. = $O(n)$

T.C. = $O(n)$

```
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  
}
```



Approach 3 'Most Optimised'

$$\text{arr} = \{ 5, 1, 3, 4, 2, 4 \}$$

arr.length = 6 \rightarrow 1 to 5 numbers

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

$$\text{sum 1 to 5} = \frac{5 * 6}{2} = 15$$

$$19 - 15 = 4$$

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

$$\text{A.S.} = O(1)$$

$$\text{T.C.} = O(n)$$

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 baar GFG print hoga $\cong \frac{n}{2}$ 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");
}
```

$$T.n.O. = 2n$$

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

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

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

```
    cout << "Khushi";
```

```
    cout << "Vineet";
```

```
    cout << "Archi";
```

```
}
```

$T.C. = O(n)$

$T.n.o = 3n$



Ques: Calculate the time complexity

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

$$T.n.O = n+m$$

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

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

```
for (int i=1; i<=200; i++){  
    |   Sout ("Somil");  
    |  
    3
```

T.n.o = 200

T.C. = $O(200) = O(1)$

T.C. = ?

Constant time

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

$$T.N.O. = m * n$$

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

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

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 << k  
      }  
    }  
  }  
}
```

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

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

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

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

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



Ques

HW: Calculate the time complexity

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

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

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

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

\vdots

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

$i=n \rightarrow \infty$

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

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

***Ques:** Calculate the time complexity

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

T.N.O = 'i' ki value jitni baar change hogi

i = 1, 2, 4, 8, 16, 32, ... n

1	2	3	4	5	6		x
						<u>total 'x' terms</u>	

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

$$x = ?$$

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

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

$$x-1 = \log_2 n$$

$$x = \log_2 n + 1$$

$$\log_2 n = \frac{\log n}{\log 2}$$

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

H.W. 1) `for(int i=n; i>=1; i/=2){`
| `cout("vinay");`
3

2) `for(int i=1; i<=n; i*=3){`
| `cout("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) {
    for(int j = 1; j<= n; j++) {
        sout("GFG");
    }
}
```

$\log n$

n

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

Fast kya hai?

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

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

↓
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");
    }
}
```

$$i=1 \rightarrow j=1$$

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

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

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

⋮

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

$$T.N.O = \underbrace{1 + 2 + 4 + 8 + \dots + n}_{n \text{ terms}}$$

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

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

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

* **Ques:** Calculate the time complexity

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

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

$$i=1 \rightarrow j=1$$

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

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

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

⋮

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

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

$$\begin{aligned}
 S+1 &= \underbrace{1+1} + 2 + 4 + 8 + 16 + \dots + n \\
 &\quad \underbrace{2+2} + 4 + 8 + 16 + \dots + n \\
 &\quad \quad \underbrace{4+4} + 8 + 16 + \dots + n \\
 &\quad \quad \quad \underbrace{8+8} + 16 + \dots + n \\
 &\quad \quad \quad \quad 16
 \end{aligned}$$

$$S+1 = 2n$$

$$S = 2n - 1$$

Ques: Calculate the time complexity

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

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

$$T.n.o = \sqrt{n}$$

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

$$i^2 \leq n$$

$$i^2 \leq n$$

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

H.W.



~~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) {
    sout("GFG");
}
```

Isko karke dikhao..



This was my best class

THANKYOU

Curties