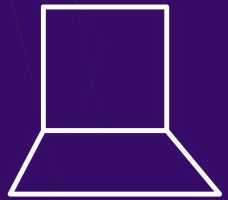
Time Complexity

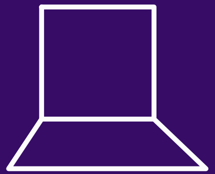
What and Why?



13^{th} gen i9

32gb ram

SSD



5^{th} gen i3

4gb ram

HDD

Ques:

Q1 : Calculate the time complexity for iterating in a loop.

```
for (int i = 0; i < n; i++) {
    System.out.println("PhysicsWallah");
}
```

→ T.n.o = n

T.C. = O(n)

Rounds → iterations

No. of iterations = no. of operations

Ques:

PW SKILLS

Q2 : Calculate the time complexity.

1)
```
for (int i = 0; i < n + 3; i ++) {
    System.out.println("PhysicsWallah");
}
```

→ T.n.o. = $n+3$

T.C. = $O(n)$

2)
```
for (int i = 0; i < n; i += 2) {
    System.out.println("PhysicsWallah");
}
```

i = 0, 2, 4, 6, ... n-1

T.n.o. = $\frac{n}{2}$

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

Approximations :

1) powers of 'n' are important

2) Highest power of n is considered

3) If there are other variables like m, they are separate.

$O(kn) \sim O(n)$

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

$O(5n+4) \sim O(n)$

$O(n^2+5) \sim O(n^2)$

$O(100\sqrt{n}+2) \sim O(\sqrt{n})$

$O(n^3+100n^2-5n) \sim O(n^3)$

$O(n^{1/3}+n^{1/2}) \sim O(n^{1/2})$

$O(n+10m) \sim O(n+m)$

Approximations :

PW SKILLS

Constant time complexity :

```
for(int i=1 ; i<=200 ; i++){
    sout("Hello");
}
```

T.n.o. = 200

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

Ques:

Q3 : Calculate the time complexity for traversing 2 arrays of size n and m.

```
int[] a = new int[n];
int[] b = new int[m];

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

for (int i = 0; i < m; i++) {
    b[i] = m - i;
}
```

$\rightarrow n \cdot 0 \Rightarrow n$

$\rightarrow n \cdot 0 \Rightarrow m$

$\rightarrow T \cdot n \cdot 0 = n + m$

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

Ques:

PW SKILLS

Q4 : Calculate the time complexity in nested loops.

n times

m times

```
for (int i = 0; i < n; i++) {
  for (int j = 0; j < m; j++) {
    System.out.print("okay");
  }
}
```

$T.n.o = n * m$

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

$i = 0 \rightarrow j = 0, 1, 2 \dots m-1$

$1 \rightarrow j = 0, 1, 2 \dots m-1$

2

3

$\vdots$

$n-1$

Note:

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

$O(n) > O(m * n)$

Ques:

Q5 : Calculate the time complexity in nested loops.

```java
for (int i = 0; i < n; i++) {
    for (int j = 0; j < i; j++) {
        System.out.print("okay");
    }
}
```

→ n times

→ i times

i		
0	→ j = α	0 times
1	→ j = 0	1 time
2	→ j = 0, 1	2 times
3	→ j = 0, 1, 2	3
.		
n-1	→ j = 0, 1, 2 ... n-2	n-1 times

$T.n.o = n*i \rightarrow \alpha$

$$T.n.o. = 0 + 1 + 2 + 3 + \dots n-1$$

$$= \frac{(n-1)(n-1+1)}{2} = \frac{n*(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} \Rightarrow O\left(\frac{n^2}{2} - \frac{n}{2}\right) \sim O(n^2)$$

In nested loops

```
for( i = 1 to n) {
    for(j=1 to m){
        for(k=1 to t){
            ____
        }
    }
}
```

$\rightarrow O(n*m*t)$

Ques:

Q6 : Calculate the time complexity for the below code snippet.

```
int c = 0;
for(int i = 1; i <= n; i*=2) {
    c++;
}
```

T.n.o $\neq$ n

T.n.o. = $x+1$

T.C. = $O(x)$

$i = 1 \quad 2 \quad 4 \quad 8 \quad 16 \; . \; . \; . \; . \; . \; 2^x$

$2^x \leftrightarrow n$

$$\underbrace{2^0 \quad 2^1 \quad 2^2 \quad 2^3 \; . \; . \quad . \; . \; . \; 2^x}_{x+1 \text{ terms}}$$

$2^x \cong n$

$$\boxed{x = \log_2 n}$$

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

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

$$O(\log_2 n) = O\left(\frac{\log n}{\log 2}\right) = O\left(\underbrace{\frac{1}{\log 2}}_{\text{Constant}} \cdot \log n\right)$$

$$\Rightarrow O(\log n)$$

Ques:

PW SKILLS

Q7 : Calculate the time complexity for the below code snippet.

```
int c = 0;
for(int i = 1; i <= n; i*=k) {
    c++;
}
```

constant

$k^x = n$

$\boxed{\log_k n = x}$

$\Rightarrow T.C. = O(\log_k n)$

$= O(\log n)$

$i = \underbrace{1, k, k^2, k^3 \dots k^x}_{x+1 \text{ terms}}$

$k^x \downarrow n$

$T.C. = O(x+1) = O(x)$

Space Complexity & Auxiliary Space

Total space used (in terms of n, m..) approximated

Extra Space used by our algorithm

Ques:

Q8 : Calculate the time and space complexity for the below code snippet.

a


```
int[] a = new int[n];

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

$\rightarrow$ T.n.o = n

T.C. = O(n)

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

Ques:

Q9 : What will be the space complexity if we just traverse without creating any array?

```
int c = 0;
for(int i = 0; i < n; i++) {
        c++;
}
int[] a  = new int[10];
```

T.C. = $O(n)$

S.C. = $O(1)$

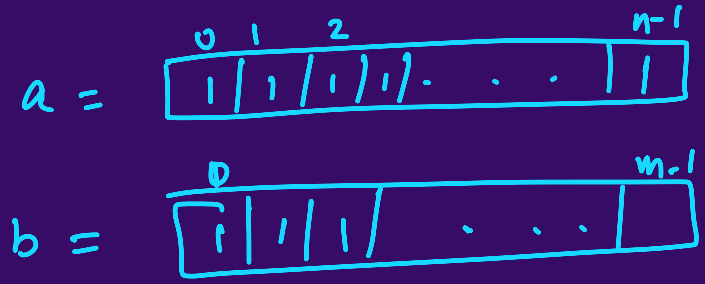
Ques:

Q10 : Calculate the space complexity for the below ~~nested~~ loop code snippet.

```
ArrayList<Integer> a = new ArrayList<>();
ArrayList<Integer> b = new ArrayList<>();
for (int i = 0; i < n; i++) {
    a.add(1);
}
for (int i = 0; i < m; i++) {
    b.add(1);
}
```

space = $n + m$

time = $n + m$



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

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

Space Complexity of creating a 2d matrix

```
int[][] a = new int[n][m];
```

total elements = $n*m$

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

Ques:

Q11 : What will be the space complexity if we create 3 arrays of the same size?

```
int[] a = new int[n];
int[] b = new int[n];
int[] c = new int[n];
for (int i = 0; i < n; i++) {
    c[i]++;
}
```

time = n

space = $n + n + n = 3n$

T.C. = $O(n)$

S.C. = $O(3n) \sim O(n)$

Ques:

Q12 : Calculate the time complexity for the following code snippet.

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

$i *= 2$

T.n.o. = ?

$i = \underbrace{1, 2, 4, 8 \ldots 2^x}_{n}$

$x += y$

$x = x + y$

$\boxed{i += i}$

$i = i + i$

$i = 2 * i$

$\boxed{i *= 2}$

$T.C. = O(x)$

$= O(\log n)$

Ques:

Q13 : Calculate the time complexity for the following code snippet.

$i = 1, 2, 4, \ldots 2^x$

$(n-1)$

$i *= 2$

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

$i = 1 \rightarrow j = 0 \quad 1$

$i = 2 \rightarrow j = 0, 1 \quad 2$

$i = 4 \rightarrow j = 0, 1, 2, 3 \quad 4$

$i = 8 \rightarrow j = 0, 1, 2, 3, 4, 5, 6, 7 \quad 8$

$\vdots$

$i = n-1 \rightarrow j = 0, 1, \ldots n-2 \quad n-1$

$T.n.O = 1 + 2 + 4 + \ldots \underset{2^x}{n-1}$

PW SKILLS

M.I

$$\boxed{2^x = n}$$

$$S = 1 + 2 + 4 + 8 \cdots 2^x$$

$$= \frac{1(2^{x+1} - 1)}{2-1} = 2^{x+1} - 1 = 2 \cdot 2^x - 1$$

$$= \boxed{2n-1}$$

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

$$\boxed{a, ar, ar^2 \cdots ar^{n-1}}$$

$\boxed{n}$

$$S = a\left(\frac{r^n - 1}{r-1}\right)$$

M-2

$$S = 1 + 2 + 4 + 8 \cdots 128$$

$$S = \underbrace{1+1} + 2 + 4 + 8 \cdots 128 - 1$$

$$S = \underbrace{2+2} + 4 + 8 + \cdots 128 - 1$$

$$S = \underbrace{4+4}_{8} + 8 + \cdots 128 - 1$$

$$S = 2 \cdot 128 - 1$$

$$\boxed{S = 2n - 1}$$

Ques:

Q14 : Calculate the time complexity for the following code snippet.

log n times

i *= 2

```
int c = 0;
for(int i = 1; i < n;  i += i) {
    for(int j = n; j >=0; j--) {
        c++;
    }
}
```

n+1 times

$$T.C. = O((n+1) * \log n)$$

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

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

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

Ques:

Q15 : Calculate the time complexity for the following code snippet.

$i * i < n \Rightarrow i^2 < n \Rightarrow \boxed{i < \sqrt{n}}$

```
int c = 0;
for(int i = 1; i * i < n;  i *= 2) {
    for(int j = 0; j < i; j++) {
        c++;
    }
}
```

$i < \sqrt{n}$

$i = 1 \rightarrow j = 0 \qquad 1$

$2 \rightarrow j = 0, 1 \qquad 2$

$4 \rightarrow j = 0, 1, 2, 3 \qquad 4$

$\vdots$

$\sqrt{n} \quad j = 0, 1, 2 .. \sqrt{n}-1 \qquad \sqrt{n}$

$T.n.o. = 1 + 2 + 4 + 8 + \cdots \sqrt{n}$

$$\text{T.n.o.} = 1 + 2 + 4 + 8 \ldots 2^x$$

$$[\sqrt{n} = 2^x]$$

$$= \frac{1(2^{x+1} - 1)}{2-1} = 2^{x+1} - 1$$

$$= 2 \cdot 2^x - 1$$

$$= 2 \cdot \sqrt{n} - 1$$

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

$\boxed{x \sim \log n}$ mistake

$$2 \cdot 2^{\log_2 n} - 1$$

$$= 2 \cdot n - 1 \quad \times \text{ wrong}$$

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

}

$i = \underbrace{1, 2, 4, 8 \ldots \sqrt{n}}_{x+1 \text{ terms}}$

$\sqrt{n} = 2^x$

T.C. = $O(x+1) = O(x)$

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

T.C. = $O(\frac{1}{2}\log n) = O(\log n)$

$2^x = \sqrt{n}$

$x = \log_2 \sqrt{n}$

PW SKILLS

Ques:

Q16 : Calculate the time complexity for the following code snippet.

```
int c = 0;
for(int i = 2; i  < n;  i *= i) {
    c++;
}
```

$i = \underbrace{2, 4, 16, 256, 65536 \ldots n}_{\text{How many terms}}$

T.n.0 = no. of values 'i' attain

$i *= i$

$i = i * i$

$\boxed{i = i^2}$

PW SKILLS

$i = 2, 4, 16, 256, 65536 \dots n$

$= 2^1, 2^2, 2^4, 2^8, 2^{16}, \dots 2^y$

$\boxed{2^y = n}$

$\boxed{y = \log n}$

$\Rightarrow \underbrace{1, 2, 4, 8, 16 \dots y}_{\text{How many terms}}$

$\underbrace{1, 2^1, 2^2, 2^3 \dots 2^x}_{x+1 \text{ terms}}$

$\boxed{y = 2^x}$

$\boxed{x = \log_2 y}$

$T.C. = O(x) = O(\log y)$

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

COLLEGE WALLAH

Ques:

Q17 : Calculate the time complexity for the following code snippet.

```
int c = 0;
for(int i = 2; i * i < n;  i *= i) {
       c++;
}
```

$i < \sqrt{n}$

$2, 4, 16, 256 \dots \sqrt{n}$

$2^{1}, 2^{2}, 2^{4}, 2^{8} \dots 2^{x}$

$\boxed{2^x = \sqrt{n}}$

$x = \log\sqrt{n}$

$1, 2, 4, 8 \dots x$

$\underbrace{1, 2, 2^2, 2^3 \dots 2^y}_{\sim y \text{ terms}}$

$\boxed{2^y = x}$

$y = \log x$

$\log(\log\sqrt{n}) = \log(\frac{1}{2}\log n)$

$= \underbrace{\log\frac{1}{2}}_{\text{constant}} + \log(\log n)$

$T.C. = O(y) = O(\log x)$

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

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

THANK YOU