Class 7 - Complexity Analysis

Saturday, April 26, 2025 10:35 AM

↳ Time and space complexity

Time complexity

```
sout ("Hello")
sout(" Hello")
```

```
sout ("Hello")
  .
  .
  .
10 ———
```

[10^8 lines of code / sec]

→ Big O Notation ↓

$\boxed{O(\sim)}$

worst case time required in terms of number of operations.

```
for (i=0; i<n; i++) {
    sout(" Hello");
}
```

→ $O(n)$

#

$10^8 \rightarrow 1$

$1 \rightarrow \frac{1}{10^8}$

$10^9 \rightarrow \frac{1}{10^8} \times 10^9 \Rightarrow 10$ sec

$10^{12} \rightarrow 10^4$ sec $\quad \frac{10^{12}}{10^8} = 10^4$

$$\frac{1000\cancel{0}}{6\cancel{0}} = \frac{500}{3} \quad \boxed{160}\ \text{min}$$

```
for( i=0 ; i<10; i++){
        for(j=0;  j<5; j++){
                sout ("Hello");
        }
}
```

O	n	
O(n)	10^8	
O(n^2)	10^4	
O(n^3)	10^2	

10,00,00,000

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

[$\sqrt{n}$ lines]

$n = 10^{\boxed{16}}$

$(\boxed{\sqrt{n}})^2 = (\boxed{10^8})^2$

fo

$(\sqrt{n})^2 = (10^8)^2$ To

$n = 10^{16}$

log

$[\log_a b]$ → $\log b_{b}$ base a

a ki power kya karein ki ans b ho

$\log_4 4 = 1$

$\log_4 64 \to 3\log_4 4 \to 3 \times 1 = 3$

$\log_2 8 \to \log_2 2^3 \Rightarrow 3 \log_2 2 = 3$

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

$10^7 \to O(n)$

$n^2 \to 10^9$

Rules of Big O

→ $O(5) = O(10) = O(20) = O(k) \Rightarrow O(1)$] constant time

→ $O(n-1) = O(n-50) = O(n-k) \Rightarrow O(n)$] linear

→ $O(n-1) = O(n-50) = O(n-k) \Rightarrow O(\underline{n})$ linear

→ $O(k \times n) = O(n)$

→ $O(n + \sqrt{n} + n^2 + \underbrace{n^4}_{\text{worst term decides the overall big O}}) \Rightarrow O(\underline{n^4})$

⊛ Time complexity of accessing an array element is $\underline{O(1)}$

Space complexity

↳ space taken by your algorithm

$O(\underline{1})$

```
{ int a;
  int b;
  int c;
  .
  .
  .
```

$O(\underline{4}) \rightarrow O(\underline{1})$

$O(\boxed{4} \times 5) = O(\underline{20}) \rightarrow O(1)$

int arr[] = new $\underline{\text{int}[n]}$; → $[O(\underline{n})]$