6/9/23

APCO
Date: 4017

Week → 3 Arrays & Time Complexity

Time & Space Complexity →

Time Complexity

↳ Amount of time taken by an algorithm to run as a function of length of input

ex→
```
cin >> N;
{ → for (int i=0; i<N; i++)
  {
    // operations
    cout << "Hello";
  }
}
```
∝ N

→ CPU operation → Define as O(N)

TC : O(N)

↓

Big O

Why to Study Complexity?

i) Resources are limited
ii) Measure algorithm to make efficient programs
iii) Asked by interviewer after every coⁿ you give.
iv) Good Engineer always thinks about complexity of the code written by him.

6/9/23

APCO
Date: 103

ex -> Algo A — Algo B -> good
-> same work karne ke liye Dono algo
But -> CPU High — CPU low
Performance — processing
↓
Bad

Space Complexity

↳ Amount of space taken by an algorithm to run as a function of length of input.

ex -> int a = 1;
int b[5];] -> O(1) -> constant

② -> int n, cin >> n;
int * b = new int [n];
↓
n ↑ — space ↑ — Space Complexity -> O(n)

Unit to Represent Complexity ->

① -> Big O : Upper bound

② -> Theta Θ : Average Case

③ -> Omega Ω : Lower bound

Date: 1/09

ex-> Search; [1|2|3|4|5|6]

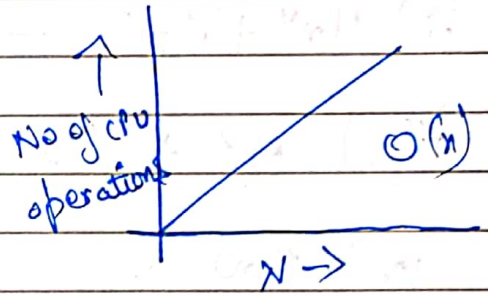
Best Case -> item found -> (1) -> $\Omega(1)/O(1)$

Worst Case -> $O(n)$

Some Big O: Complexities

(1) -> Constant time -> $O(1)$

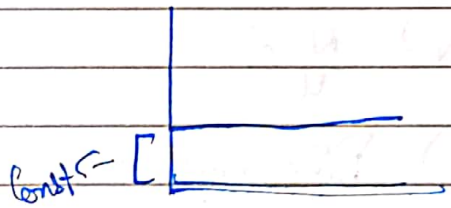
(2) -> Linear time -> $O(n)$

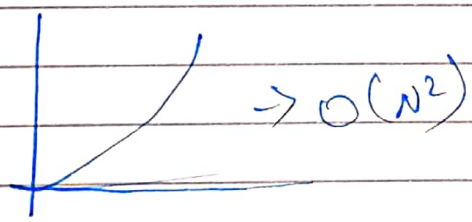


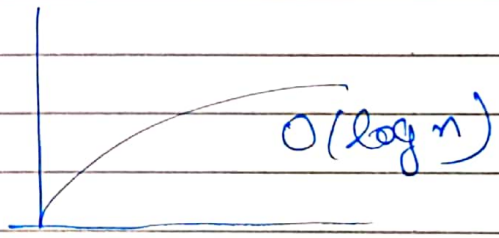
(3) -> Logarithmic time -> $O(\log N)$

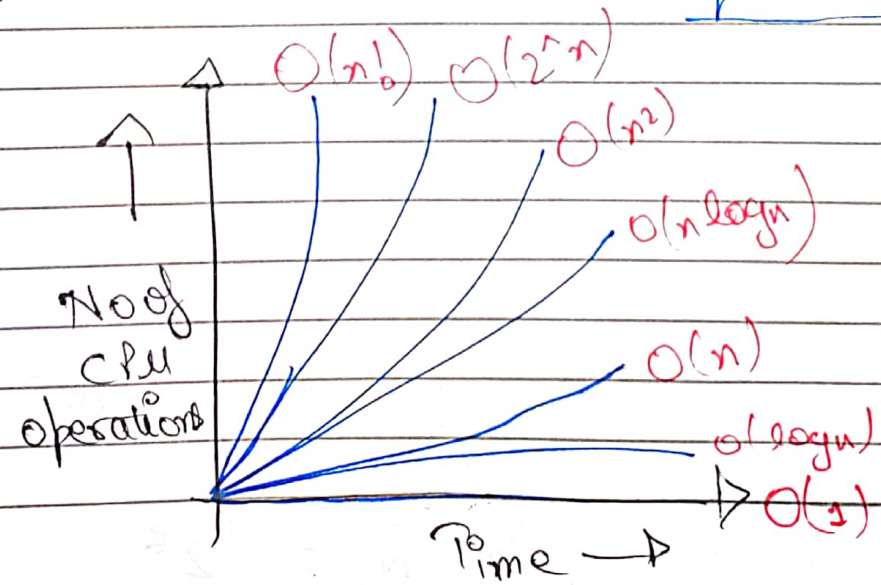
(4) -> Quadratic time -> $O(N^2)$

(5) -> cubic time -> $O(N^3)$









Best (least)

ARCO
Date: 1/10

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

worst

ex: $f(n) = 2n^2 + 3n \Rightarrow O(2n^2) \Rightarrow O(n^2)$

constant

& as compare to n^2, 2 does not matter

i) $4n^4 + 3n^3 \Rightarrow O(4n^4) \Rightarrow O(n^4)$

ii) $N^2 + \log N \Rightarrow O(N^2)$

iii) $200 \rightarrow O(1)$

iv) $\frac{N}{4} \rightarrow O(N)$

v) Binary Search $\rightarrow O(\log n)$ find 6

1 2 3 4 5 6 7

4 < 6 → right

5 6 7 → Midly

① → N → Search space

② → N/2

③ → N/4 —— 1

$N \to N/2 \to \frac{N}{4} \cdots \frac{N}{2^k} = 1$

$N/2^k = 1$

$\Rightarrow N = 2^k$

APCO
Date: 7/1/23

6/9/23

$\Rightarrow \log_2 N = 1 \cdot \log_2 2^k \Rightarrow \log_2 N = k \log_2 2$

$k = \log_2 N \rightarrow TC \rightarrow O(\log N)$

Questions → i)
```
int main(){
    int a=0, b=0, n, m;
    cin >> n >> m;
    for(int i=0; i<n; i++){        → O(n)
    }
    for(int i=0; i<m; i++){        → O(m)
    }
    return 0
}
```
$T.C = O(n) + O(m)$

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

ii)
```
int main(){
    int a=0; b=0, n;
    cin >> n;
    for(int i=0; i<n; i++){        → n tak chalegaa
        for(int j=n; j>i; j--){
            cout << "Hi" << "\n";
        }
    }
    return 0;
}
```
i = 0 → n to 0 → n hi hua

i = 1 → n to 1

i = 2

So, worst case n to 0 hi chalgaa

& Nested loop, so

$TC \rightarrow O(n^2)$