Topics to discuss

Running Time Analysis

How to find Time Complexity.

YouTube @StartPracticing

Eg.1

```
int a = 20; — 1 unit time
int b = 30; — 1 unit time
int sum = a+b; — 1
print(sum); —— 1 unit time
```

User independent

Algorithm

- Iterative

```
For(i → n)
{
  ...
}
```

- Recursive

```
A(n)
{
  ...
  A(n/2)
}
```

How much time it will take?

Total time = 4 unit of time

Time complexity = O(1) or O(c)

YouTube @StartPracticing

Eg-2 :

```
Algorithm sum (A[], n)          input
{
  sum = 0;  ———————  1 × 1 unit time
  for (i = 1 ; i ≤ n ; i++)  ——— (n+1) × 1 unit time
  {
    sum = sum + A[i];  ———————  n × 1
  }
}
```

Total time = $1 + (n+1) + n$

$f(n) = 2n + 1$

$f(n) = O(n)$

$f(n) \leq c \cdot g(n)$

YouTube @StartPracticing

Eg-3: Algorithm Sum (A[][], B[][], n)

```
{
  for (i=1 ; i<=n ; i++)          ——— (n+1) x 1
  {
    for (j=1 ; j<=n ; j++)        ——— n x (n+1)
    {
      C[i][j] = A[i][j] + B[i][j]; ——— n x n
    }
  }
}
```

Total time $= (n+1) + n(n+1) + n^2$

$= n+1+n^2+n+n^2$

$f(n) = 2n^2 + 2n + 1$

Time Complexity $= O(n^2)$

YouTube @StartPracticing

Eg-4:

```
Algorithm Multiply (A[][] , B[][] , n)
{
    for (i=1 ; i<=n ; i++)                    ——— n+1
    {
        for (j=1 ; j<=n ; j++)                ——— n x n+1
        {
            C[i][j] = 0;                      ——— n x n
            for (k=1 ; k<=n ; k++)            ——— n x n x n+1
            {
                C[i][j] = C[i][j] + A[i][k] * B[k][j];   ——— n x n x n
            }
        }
    }
}
```

$$\text{Total time} = n+1 + n(n+1) + n^2 + n^2(n+1) + n^3$$

$$= n+1 + n^2 + n + n^2 + n^3 + n^2 + n^3$$

$$f(n) = 2n^3 + 3n^2 + 2n + 1$$

$$\boxed{f(n) = O(n^3)}$$

YouTube @StartPracticing

Follow Now

Start Practicing

i._am._arfin

Arfin Parween

YouTube @startPracticing