**Basics of Time & Space Complexity**

* **Time Complexity:**

**🡪**Rate at which the time taken increases with respect to the input size

🡪Time Complexity != Time Taken

🡪It can’t be represented in terms of seconds. Instead it is represented in terms of **Big O Notation**

* **Three Rules while calculating Time Complexity:-**

1. Always calculate TC in terms of worst case scenario

2. Avoid constants

3. Avoid lower values

* **Just for the knowledge remember this:-**

🡪Big-Oh (O) Complexity 🡪 Worst Case [Upper Bound]

🡪Theta (θ) Complexity 🡪 Average Case

🡪Omega (Ω) Complexity 🡪 Best Case [Lower Bound]

* **Space Complexity:-**

**🡪Auxiliary Space:** Space that you take to solve the problem

**🡪Input Space**: Space that you take to store the input

* **GCD/HCF:-**

**Euclidean Algorithm**: gcd(a, b) = gcd(a % b, b), where a > b

**Example**: let a = 20, b = 15

gcd(20, 15) = gcd(5, 15)

gcd(15, 5) = gcd(0, 5)

So, when any of the no. becomes 0, the other no. will be the gcd/hcf

* **Recursion:-**

🡪When a function calls itself until a specified condition is met.

🡪**Segmentation Fault (Stack Overflow):** When the numerous functions calls waiting in the stack memory due to recursion