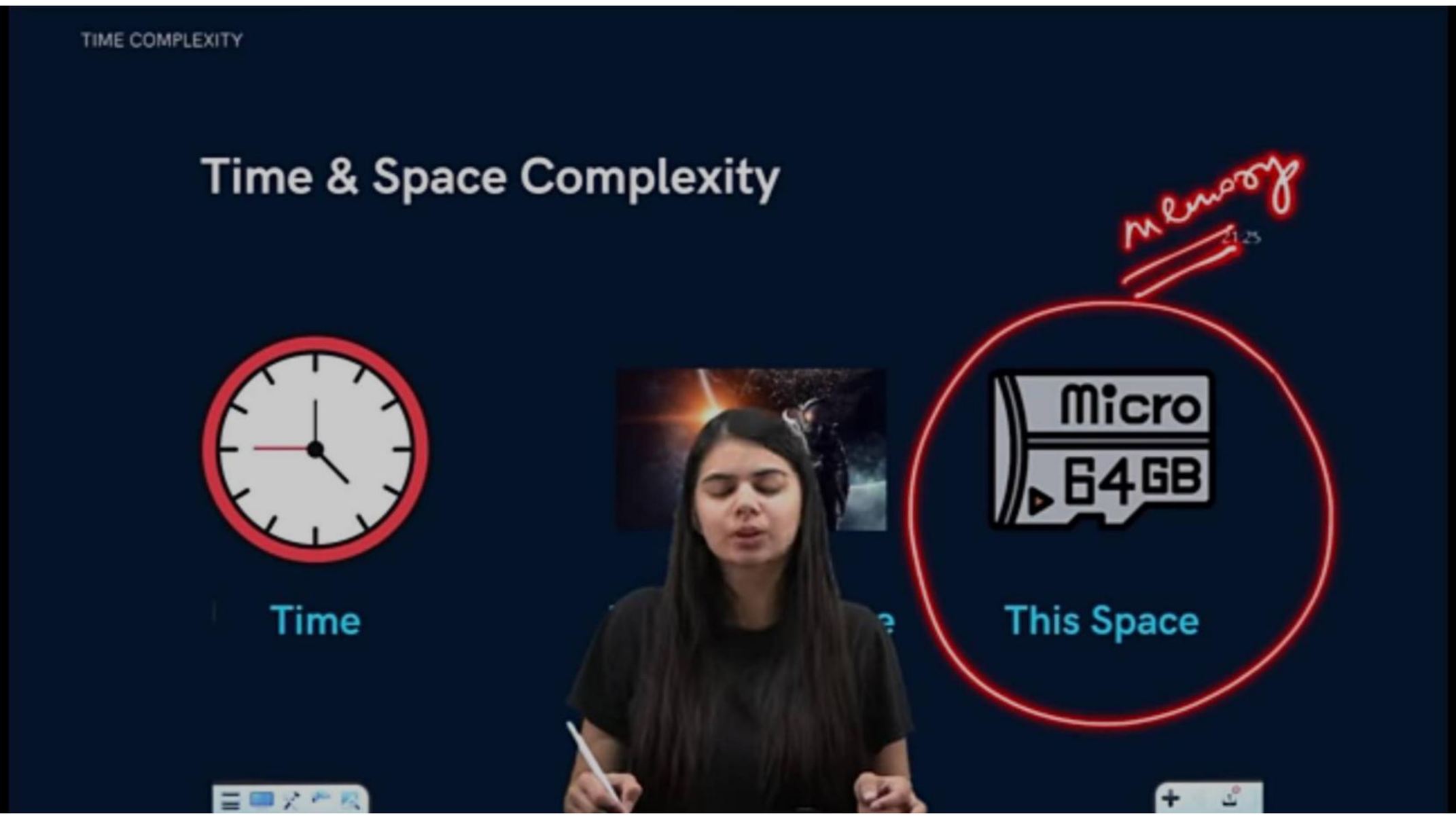
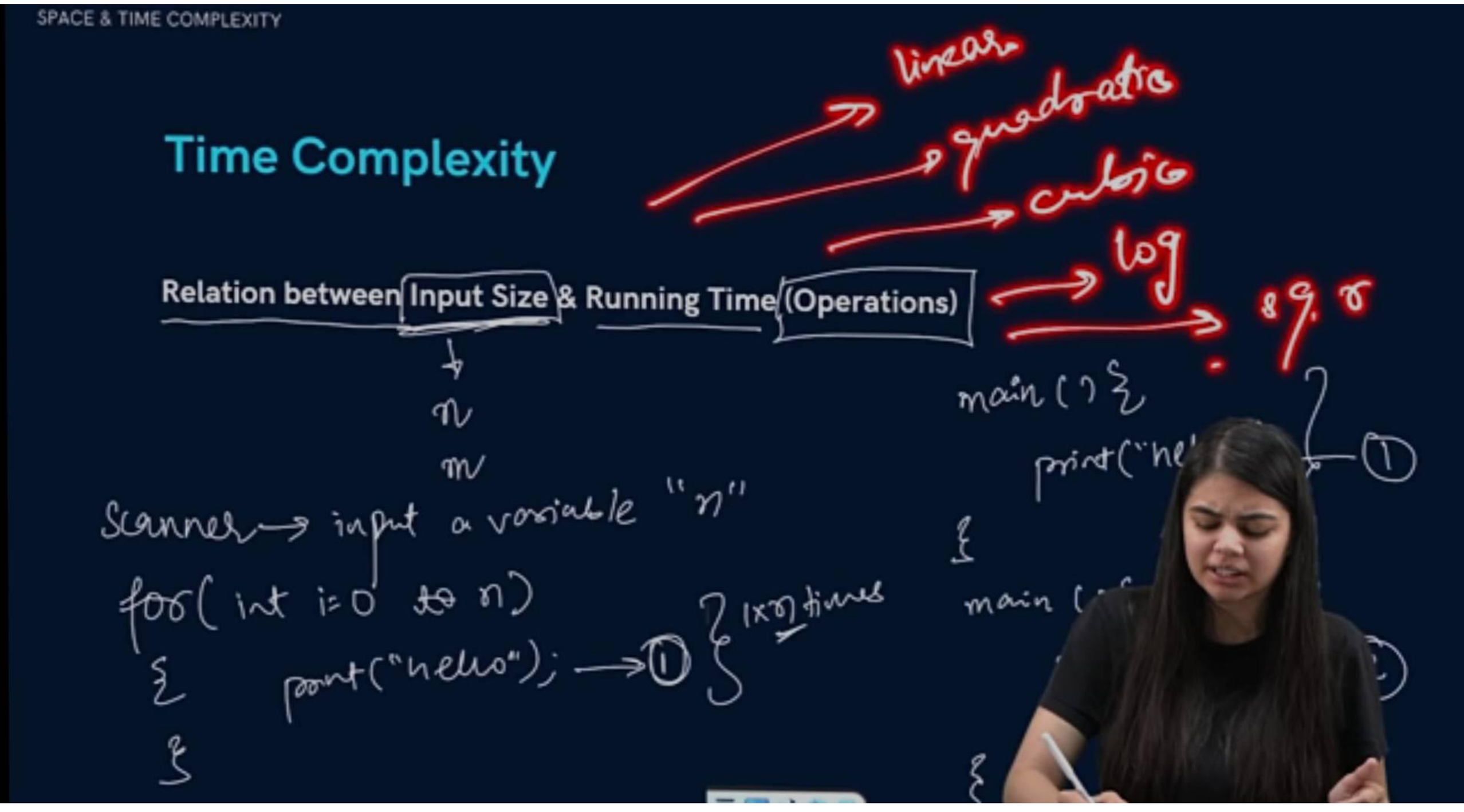




SPACE & TIME COMPLEXITY

1 Time Complexity

```
public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    for(int i=0; i<n; i++) {
        System.out.println("hello");
    }
}
```

input n

n

time

n times

SPACE & TIME COMPLEXITY

1 Time Complexity

```
public static void main(String args[]) {
Scanner sc = new Scanner(System.in);
int n = sc.nextInt(); }

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

input n → n

time complexity ∝ input n

→ n times

SPACE & TIME COMPLEXITY

Numbers : {1, 2, 3, 4, 5}

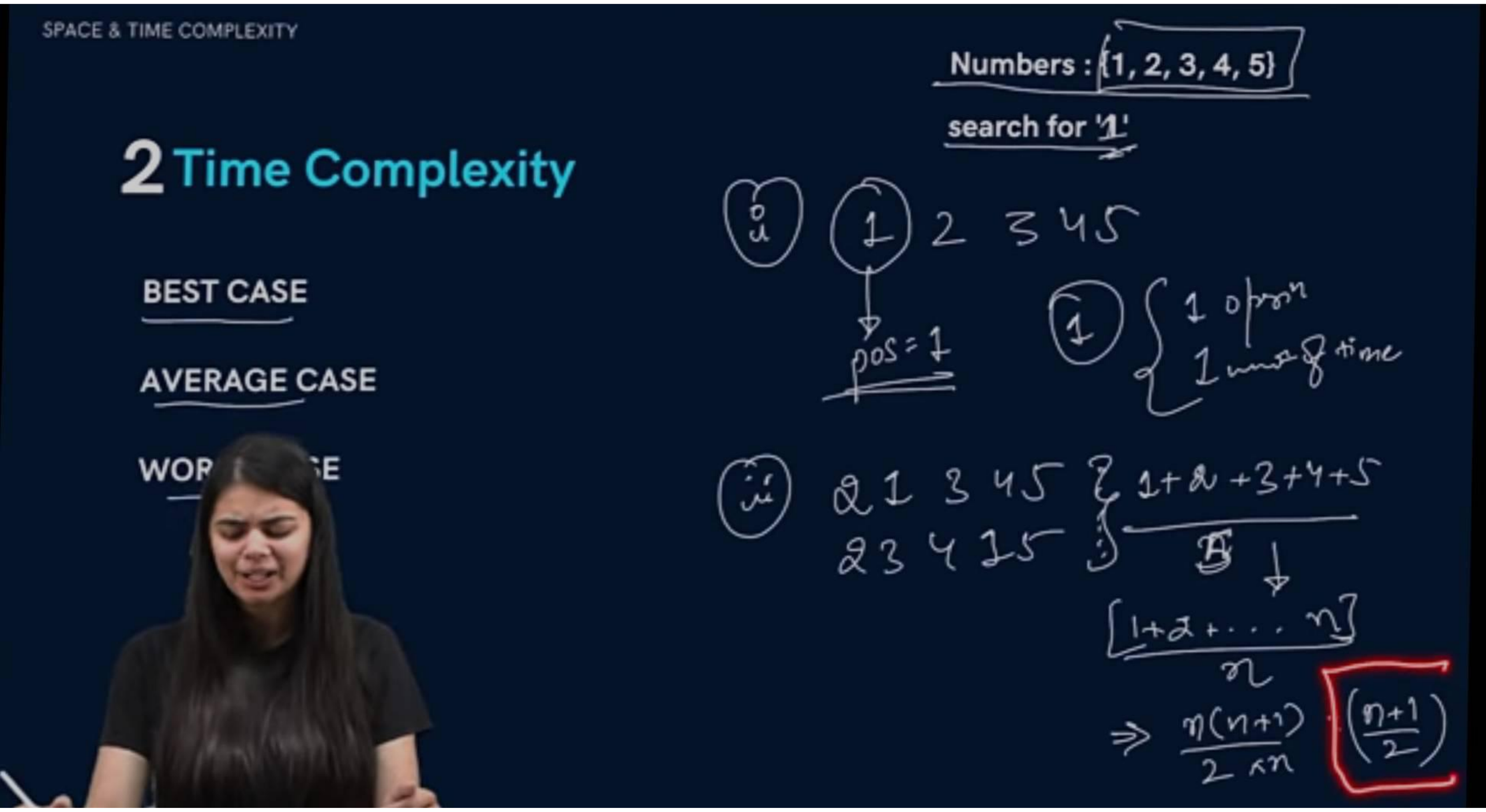
search for '1'

2 Time Complexity

BEST CASE

AVERAGE CASE

WORST CASE



2 Time Complexity

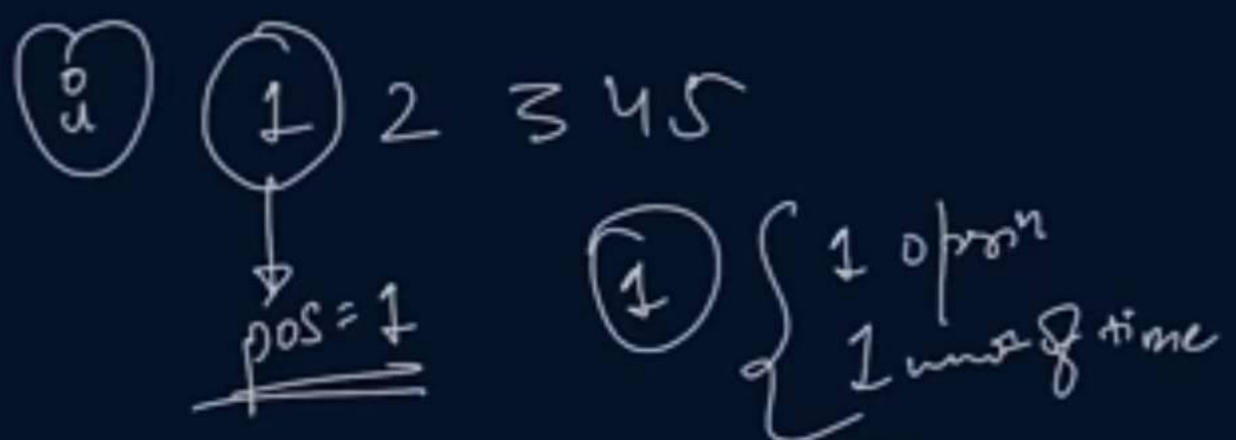
BEST CASE → $\Omega(1)$

AVERAGE CASE → $\Theta(\frac{n+1}{2})$

WORST CASE → $O(n)$

Numbers : {1, 2, 3, 4, 5}

search for '1'



SPACE & TIME COMPLEXITY

3 Time Complexity

```
public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    for(int i=0; i<n; i++) {
        for(int j=0; j<n; j++) {
            System.out.println("hello");
        }
    }
}
```

SPACE & TIME COMPLEXITY

Dynamic
1

7 Time Complexity

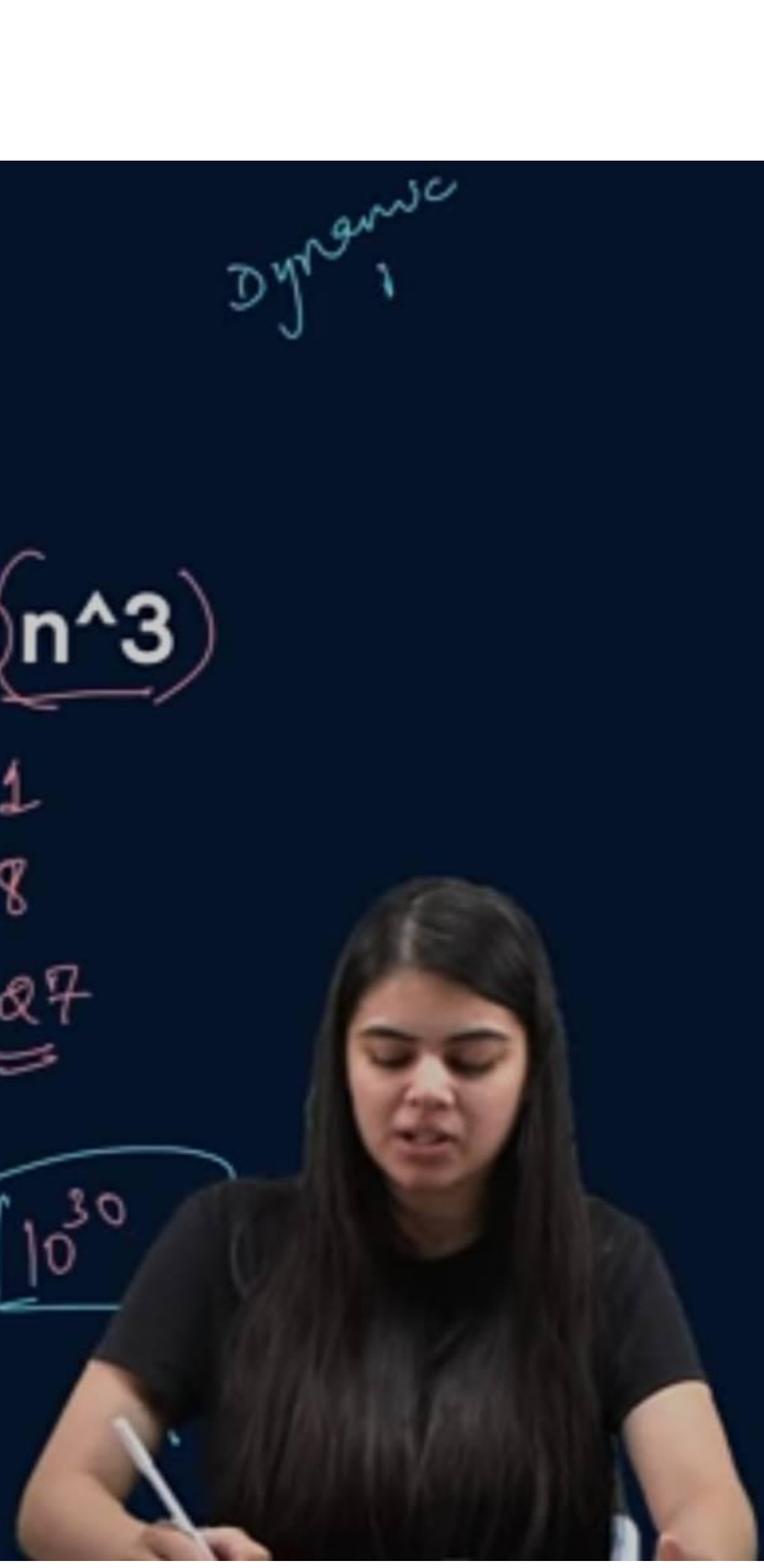
Compare : O(n) O(n^2) O(n^3)

	O(n)	O(n^2)	O(n^3)
n = 1	1	1	1
n = 2	2	4	8
n = 3	3	9	27

10^5

10^{10}

10^{30}



SPACE & TIME COMPLEXITY

1 Space Complexity

```
public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();

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