Your Ultimate Guide To Landing Top AI roles

2.3.2 Time Complexity - Iterative Programs

→ Using time complexity, we can compare multiple Algorithm/Program in terms of execution time

→ There are 2 types of Algorithms

① Iterative → Loops

```
def sum_nums(data):
    sum_data = 0
    for val in data:
        sum_data += val
    return sum_data
```

② Recursive → Recursion - A function calling itself.

```
def calc_factorial(n):
    if n==0:
        return 1
    else:
        return n*fact(n-1)
```

←

Iterative Version

```
def calc_fact(n):
    fact = 1
    for i in range(n+1):
        fact *= i
    return fact
```

→ For today's lecture, we will focus on time Complexity of iterative programs.

→ In order to analyze iterative program, we have to count no of times loop gets executed

Ex: Given n, print square of all no from [1, n]

```
def helper(n):
    for i in range(1, n+1):
        print(i*i)
```

Time Complexity = no of loop iterations

$T(n) = n \text{ times} = O(n)$

→ NOTE : If any program/Algorithm don't contain loops or recursion it means that there is no dependency of running time on Input Size (n). For such program time Complexity = O(1)

> O(1) means time taken by the program to execute is independent of n.

Ex:

```
def function_name():
    print("Hello ! good morning")
    a = 5000
    b = "Hello Sanjeev"
    print(b)
```

→ $T(n) = O(1)$

Ex:-
```
for i in range(n):
    for j in range(n):
        print("Decode AiML")
```

Ex:-
```
i, s = 1, 1
while s<=n:
    i += 1
    s += i
    print("Decode AiML")
```

Ex:-
```
for i in range(n):
    for j in range(i):
        print("Decode AiML")
```

Ex:-
```
for i in range(n):
    for j in range(i):
        for k in range(100):
            print("Decode AiML")
```

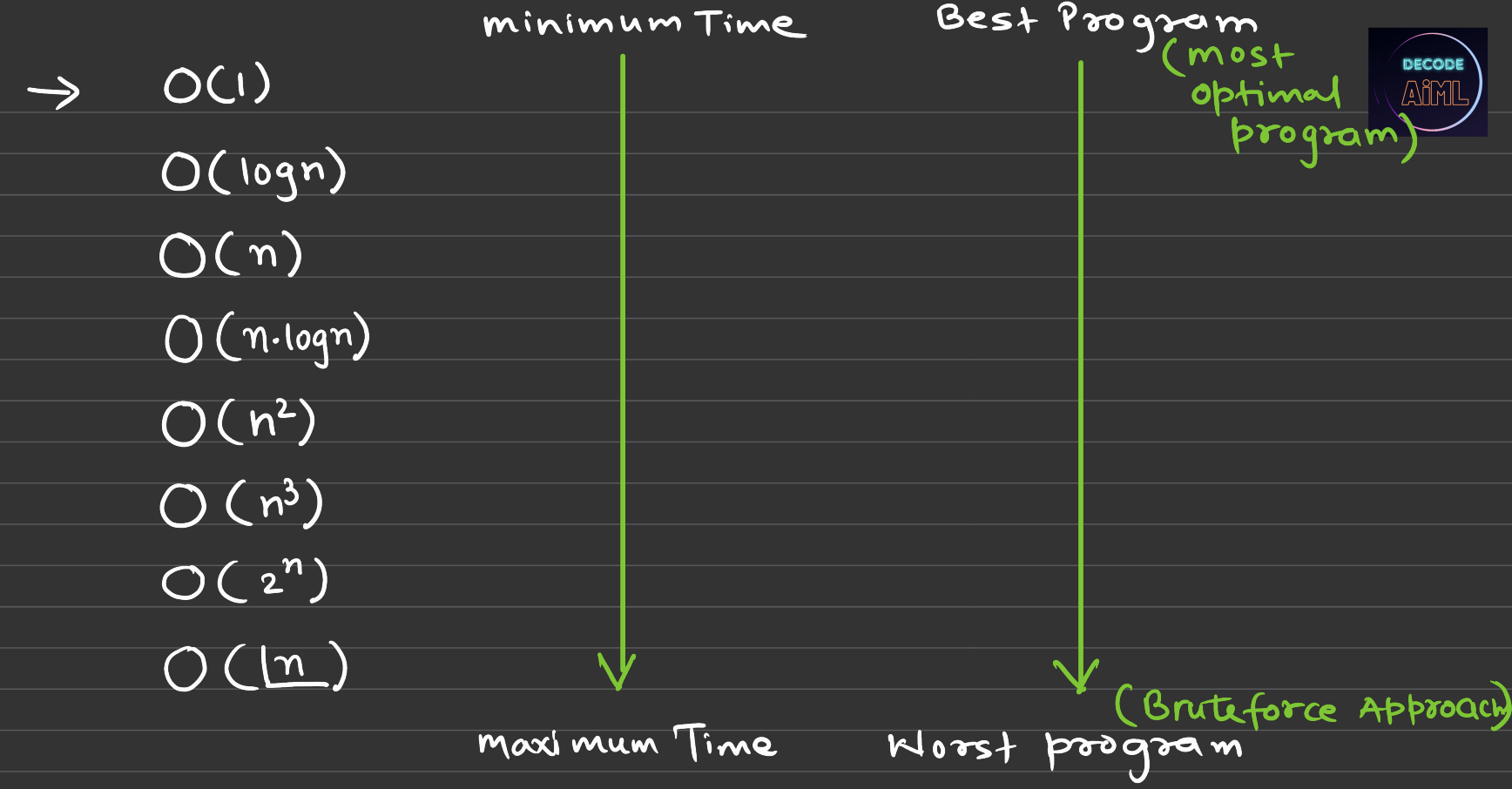
Ex:-
```
i=1
while i<=n:
    i = i*2
    print("Decode AiML")
```

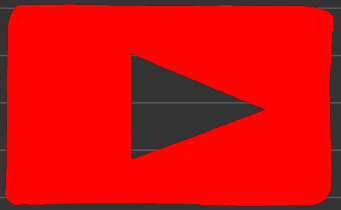
Ex: i, J, K = 0, 0, 0

```
for i in range(n/2, n+1):
    J = 1
    while J <= n:
        J = 2 * J
        K = 1
        while K <= n:
            K = K * 2
            print("Decode AiML")
```

Ex:- while n>1:
    n = n/2
    print("Decode AiML")

Ex:- for i in range(n):
    for j in range(0,n,i):
        print("Decode AiML")


minimum Time
Best Program
(most optimal program)
DECODE AiML
O(1)
O(logn)
O(n)
O(n.logn)
O(n²)
O(n³)
O(2ⁿ)
O(n!)
maximum Time
Worst program
(Bruteforce Approach)

Like  Subscribe