Analysis of Algorithms (Background)

Sum of n natural numbers: (Make a function)

Input: n = 3

Output: 6 // 1 + 2 + 3

Input: n = 5

Output: 15 // 1 + 2 + 3 + 4 + 5

Approach 1:

```
def func1(n):
        return n * (n + 1)/2
```

Approach 2:

```
def func2(n):
        sum = 0
        for i in range(1, n + 1):
                sum += i
        return sum
```

Approach 3:

```
def func3(n):
        sum = 0
        for i in range(1, n + 1):
                for j in range(1, i + 1):
                        sum += 1
        return sum
```

Order of Growths:

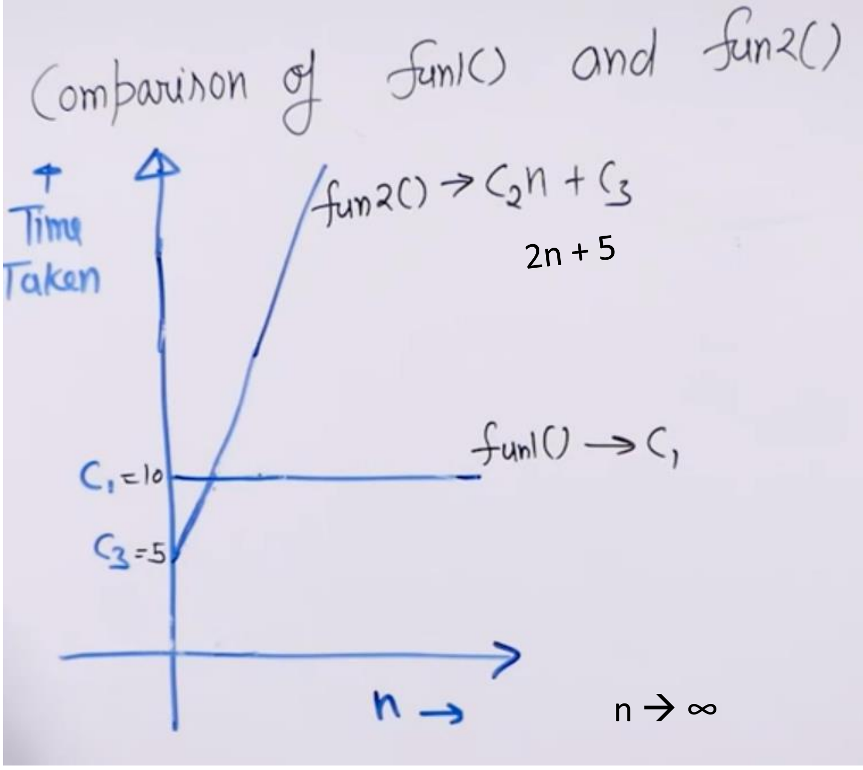
func1() → c_1

func2() → $c_2n + c_3$

func3() → $c_4n^2 + c_5n + c_6$

Asymptotic Analysis: (Theoretical Analysis)

- No dependency on machine, programming language etc.
- We do not have to implement all ideas/algorithm
- Asymptotic analysis is about measuring order of growth in terms of input size



2n + 5

n → ∞

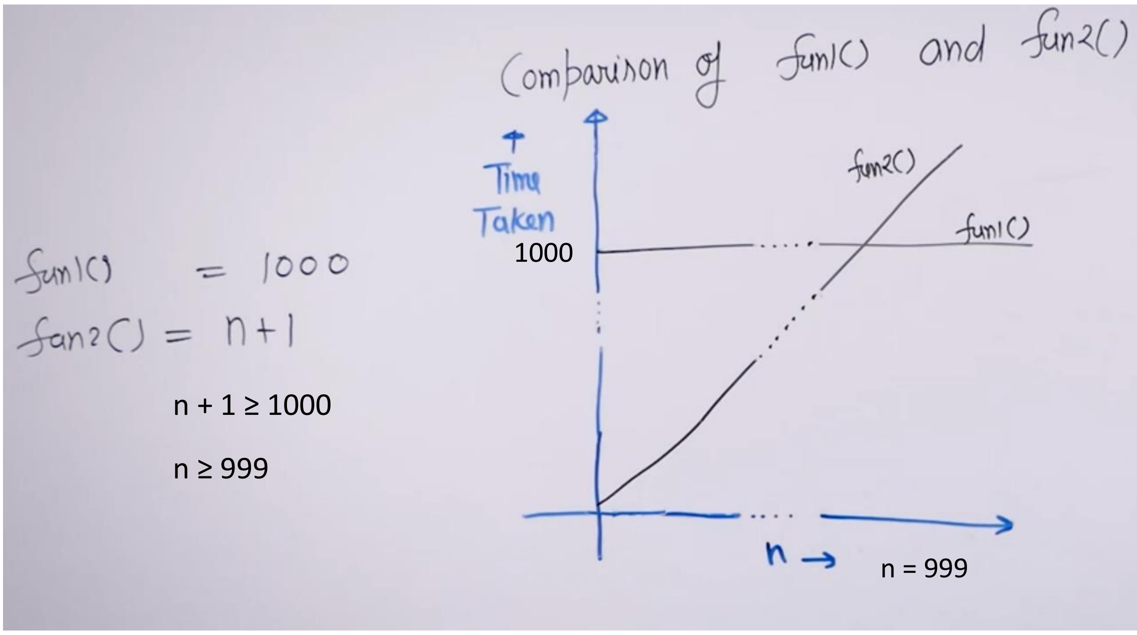
$2n + 5 \geq 10$

$n \geq 2.5$

$n \geq 3$

$n \geq 0$

fun1(), fun2() ≥ 0



fun1() = 1000

fan2() = n+1

1000

$n + 1 \geq 1000$

$n \geq 999$

n = 999