Lec-11

07/12/25 12:45 PM

Time & Space Complexity

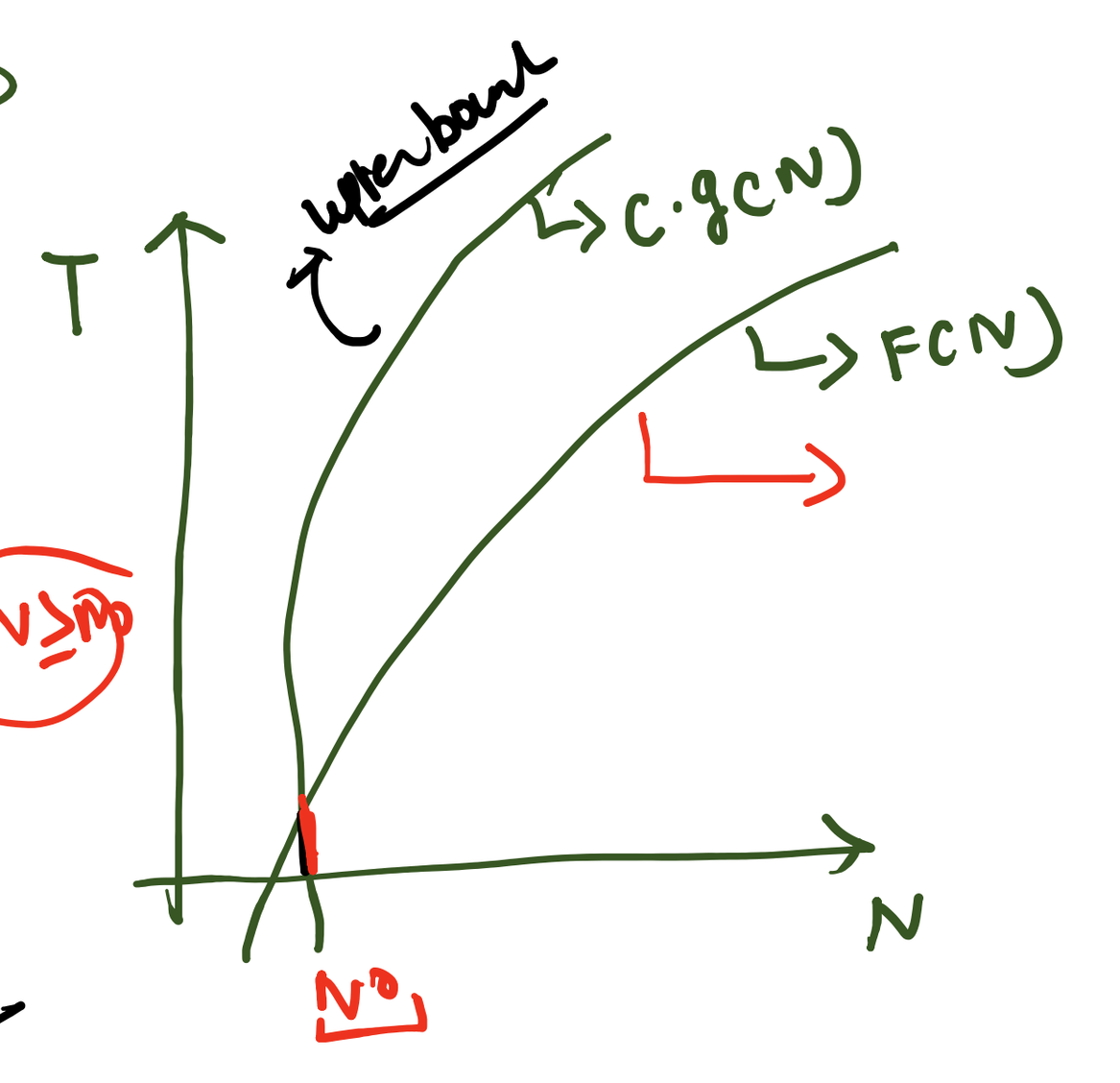
asymptotic notation vs Experimental Way

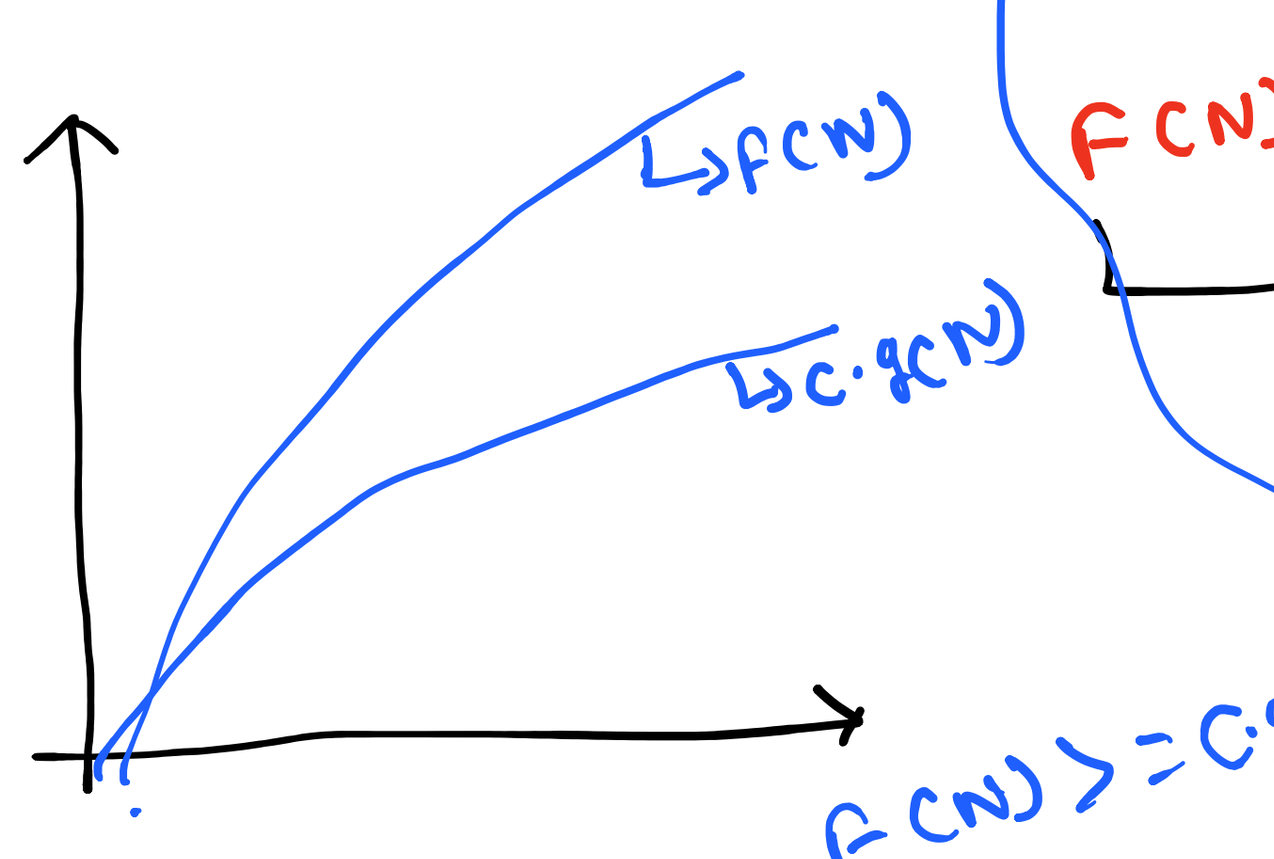
Collex → asymptotic notation; Experimental Way → Chem, Phy; System, System depend

T.C & S.C → Algo → {Time, Space}; Time ↓ Space ↑; Space ↓ Time ↑

asymptotic notation → inputsize

- worst case
- Avg case
- best case





F(N) ≤ C.g(N) → O(g(n)) Big-oh, N ≥ N0

F(N) >= C.g(N) → Ω(g(n)), N ≥ N0

$T(n) = 3n^2 + 7N + 5$ → n^2

$T(n) = 3n^2 + 7N + 5$

$g(n) = n^2$

$c = 15$

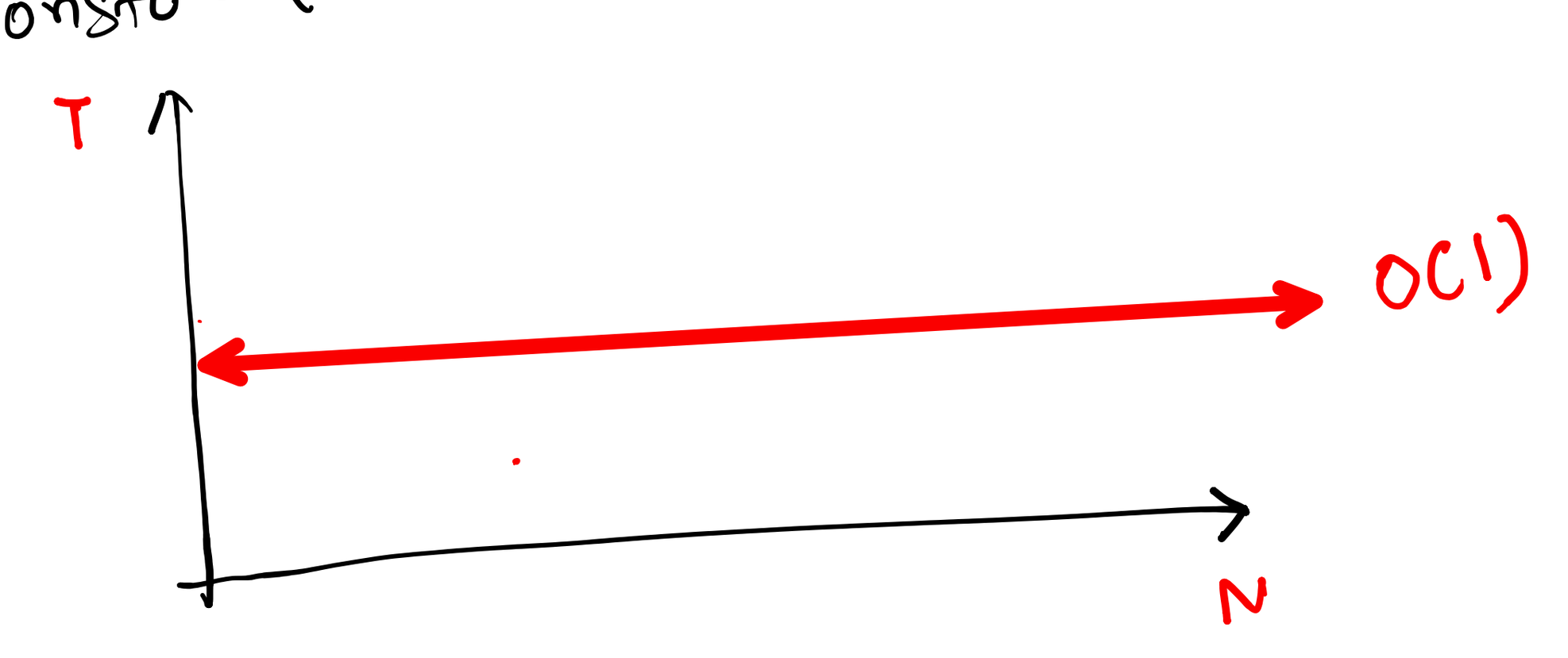
$3n^2 + 7N + 5 \le 15n^2$

$T(n) = 3n^4 + 2^n + \log(n) + 7$ → 2^n

$T(n) = 3N^3 + 7N^4 + (e^n \log(n))^n + 2^n + 7$

n^c, S/S o(n)

Constant time



O(1)

```java
public static int Search(int[] arr, int item) {
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == item) {
            return i;
        }

    }
    return -1;
}
```

+1

i=0 i=1 i=2 ...... i=n-1

1 1 1 1

T(n) = N+1 → O(N)

```java
public static int Maximum1(int[] arr) {
    // TODO Auto-generated method stub
    int max = Integer.MIN_VALUE;// -2^31
    for (int i = 0; i < arr.length; i++) {
        max = Math.max(arr[i], max);
    }
    return max;
}
```

n, 1

i=0 i=1 i=2 i=3

n

T(n) = n+1 +1 = n+2 → α n

```java
public static int Search(int[] arr, int item) {
    // TODO Auto-generated method stub
    int lo = 0;
    int hi = arr.length - 1;
    while (lo <= hi) {
        int mid = (lo + hi) / 2;
        if (arr[mid] == item) {
            return mid;
        } else if (arr[mid] > item) {
            hi = mid - 1;
        } else {
            lo = mid + 1;
        }
    }
    return -1;
}
```

$a^b = c$, $b = \log_a c$

T(n) = K ?

N → 1

N/2 → 1

N/2² → 1

N/2³ → 1

N/2⁴ → 1

⋮

K { N/2^K → 1

$\frac{N}{2^K} = 1$, $2^K = N$, $K = \log_2 N$

while(n>0){ n=n/2 }

```java
while (i <= n) {
    System.out.println("Hey");
    i *= 2;
}
```

i=1 i=2 i=2² i=2³ i=2⁴ --- i=2^K → N

1 1 1 1 1 1

T.C = K

$2^K = N$, $K = \log_2(n)$

while(n>0){ n=n/2 } | i=1 while(i<=n){ i=i*2 }

```java
for (i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        System.out.println("hey");
    }
}
```

n×n×1 = n²

T.C — Same — T.C

```java
for (i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        System.out.println("hey");
    }
}
```

i=1 i=2 i=3 ---- i=N

N N N N

T(n) = n²

```java
for (i = 1; i * i <= n; i++) {
    System.out.println("hey");
}
```

$i^2 \le N$

$i \le \sqrt{N}$

```java
for (i = 1; i <= n; i++) {
    for (int j = 1; j <= i * i; j++) {
        for ( k = 1; k <= n / 2; k++) {
            System.out.println("hey");
        }
    }
}
```

i=1 i=2 i=3 i=4 --- i=N

N/2, 2²×N/2, 3²×N/2, 4²×N/2, N²×N/2

$\frac{N}{2}[1 + 2^2 + 3^2 + 4^2 + \dots + N^2]$

$\frac{N}{2} \times \left[\frac{N \times (N+1) \times (2N+1)}{6}\right]$

```java
for (i = n / 2; i <= n; i++) {
    for (int j = 1; j <= n / 2; j++) {
        for ( k = 1; k <= n; k = k * 2) {
            System.out.println("hey");
        }
    }
}
```

$\frac{N}{2} \times \frac{N}{2} \times \log(N)$

```java
for (i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j += i) {
        System.out.println("hey");
    }
}
```

N×log(N)

i=1 i=2 i=3 i=4 ---- i=N

N N/2 N/3 N/4 N/N

$N[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}]$ → log(N)

12th → log N