Lec-12

22/03/25 9:44 AM

5min

Space

Time and Space ?

Soln

Time

Time Complexity asymptotic notation vs ~~Experimental Way~~

program Run होने [illegible] Time

maximum submissions

2 loop

System × System dependent

Hard

Asymptotic Notation → Time complexity

System independent

(ii) × {Best case}

(iii) × {Avg case}

(i) {Worst case} ✓

G.T

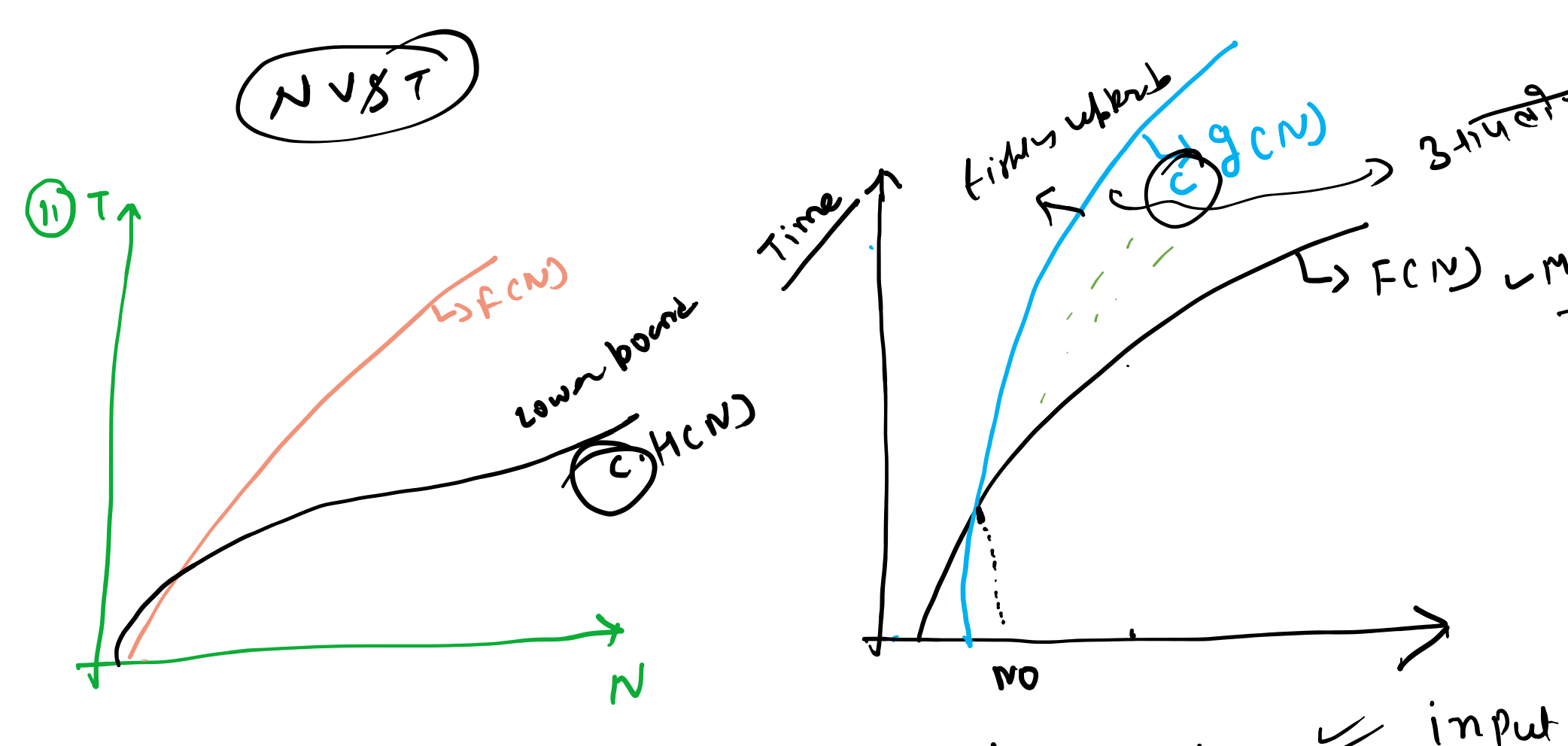
20 min

40 min

2 hours

best

N vs T



उपयोगी का code

F(N) upper code है

(i) Math ×

(ii) →

where c is constant

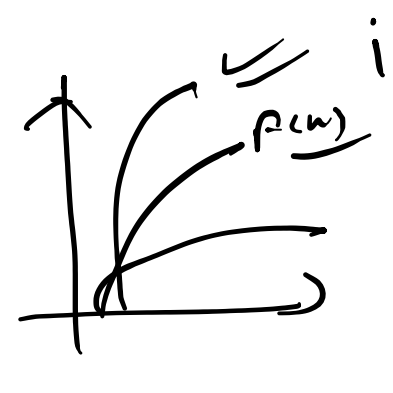
F(N) <= C·g(N)

O(g(N))

big-oh

F(N) ↪ Ω(H(N))

F(N) >= C·H(N)



input size

θ U

$F(N) <= C \cdot g(N)$

$F(N) = 2n^2 + 5n + 6$

$C\,g(N)$

$2n^2 + 5n + 6 <= 13N^2$ $N >= 1$

$O(N^2)$

(1) pnd

$C = 13$

$g(n) = n^2$

$\lim_{n \to \infty}$

$13n^2$

$3 \times \infty = \infty$

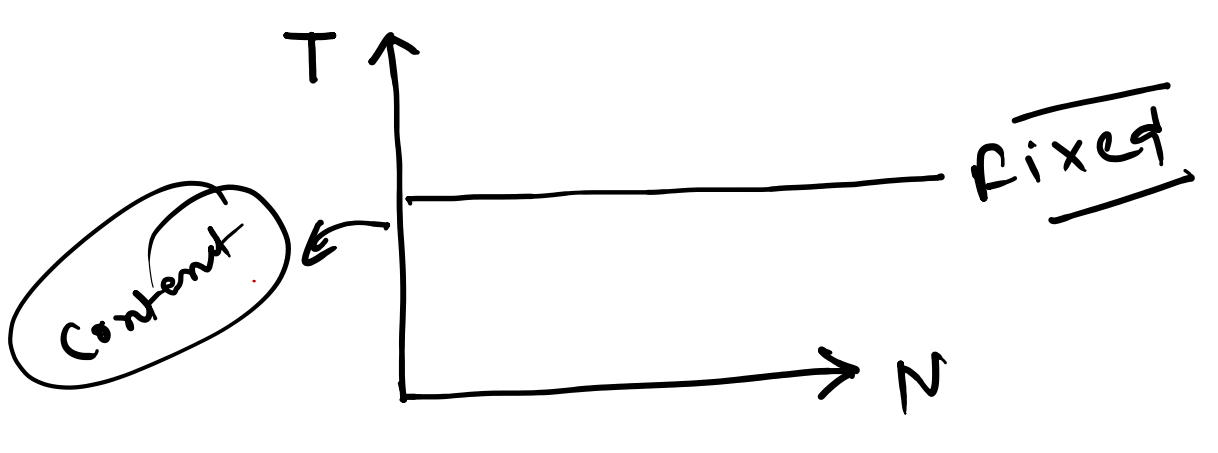
$f(n) = 7N^2 + 5(2^n \log(N)) + n^3 + \log(n) + 4$

$n^?$

T.C $2^N \times \log(N)$

(ii) constant time

O(1)



```
public static int Search(int[] arr, int item) {
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == item) {
            return i;
        }
    }
    return -1;
}
```

O(1)

1

x=5

i=0 i=1 i=2 i=3 ---- i=n-1

1 1 1 1 1

T(N) = N+1

O(n)

```
public static int Maximum1(int[] arr) {
    int max = Integer.MIN_VALUE;//-2^31
    for (int i = 0; i < arr.length; i++) {
        max=Math.max(max, arr[i]);//O(1)
    }
    return max;
}
```

O(1)

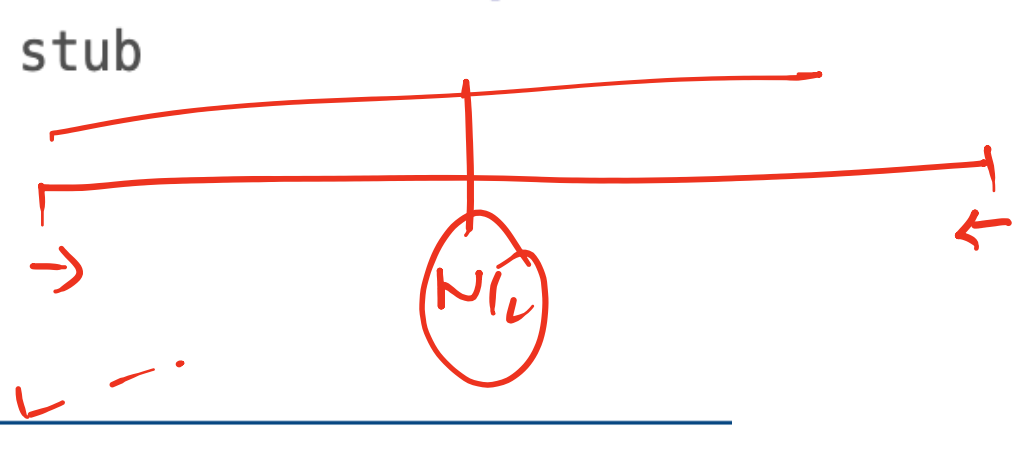
n

n+2

```
public static void Reverse(int[] arr, int i, int j) {
    // TODO Auto-generated method stub

    while (i < j) {
        int t = arr[i];
        arr[i] = arr[j];
        arr[j] = t;
        i++;
        j--;
    }

}
```



1 + 2 ---

$\frac{n}{2} = O(n)$

```
public static int Search(int[] arr, int item) {
    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) {
            lo = mid + 1;
        } else {
            hi = mid - 1;
        }
    }
    return -1;
}
```

O(1)

$\frac{N}{2^k} = 1$

k+1

O(k)

$\log_2 N$

1 ——— N

$N \to 1$

$\frac{N}{2} \to 1$

$\frac{N}{2^2} \to 1$

$\frac{N}{2^3} \to 1$

$\frac{N}{2^4} \to 1$

$\frac{N}{2^5} \to 1$

$\frac{N}{2^k} \to 1$

1024 → 512 → 256 → 128 → 64 → 32 → 16 → 8 → 4 → ...

$\frac{N}{2^k} = 1$

$2^k = N$

$K = \log_2 N$

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

i=1 i=2 i=2^2 i=2^3 --- i=2^k

1 1 1 1 1

log n

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

n

n

(1)

$n \times n + 1 = n^2$

i=1 i=2 i=3 -- i=n

N N N N

n^2

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

$i^2 <= N$

$i <= \sqrt{N}$

```
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");
            }
        }
    }
```

2×2=4

j=1 j=2 j=3 -

$\frac{N}{2}$ $\frac{N}{2}$ n

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

$\frac{N}{2}$ $2^2 \times \frac{N}{2}$ $3^2 \times \frac{N}{2}$ $4^2 \times \frac{N}{2}$ $N^2 \times \frac{N}{2}$

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

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

$O(n^4)$

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

$\log(N) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$

i=1 i=2 i=3 i=4 ---- i=n

N $\frac{N}{2}$ $\frac{N}{3}$ $\frac{N}{4}$ $\frac{N}{n}$

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

log(N)

H.P

log(n) ✓

e^n ✓

2^n

n×log(n)

$\frac{n \times (n+1)}{2} - N$

$\frac{n \times (n-1)}{2} = n^2$

n=5

4+3+2+1

$\frac{n^2+n}{2} - n$

$\frac{n^2+n-2n}{2}$

$\frac{n^2-n}{2}$

$\frac{n(n-1)}{2}$

```
public static void Sort(int[] arr) {
    for (int turn = 1; turn < arr.length; turn++) {
        for (int i = 0; i < arr.length - turn; i++) {
            if (arr[i] > arr[i + 1]) {
                int temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
            }

        }
    }
}
```

4 ✓

3 ✓

2 ✓

1 ✓

Constraint

$n <= 10^5$

n^2

10^{10}

1 sec → 10^8 to 10^9

10^7 to 10^8

$n \times \sqrt{n}$