Lec-11
06/04/24 12:48 PM

Time and space Complexity

(i) (ii)

experimental way and asymptotic notation

(i) (ii)

(i) Best → best
(ii) Worst
(iii) Avg → Avg

Worst case

$f(n) \to g(n)$

$f(n) \le c \cdot g(n)$

$f(n) \le c \cdot g(n)$ $\quad n \ge n_0$

$f(n) = O(g(n))$

$f(n) = 3n^2 + 4$

$T = 3n^2 + 4n + 7$

$T = 3n^2 + 4n + 7$

$c = 14$ $\quad n \ge 1$

$g(n) = n^2$

$3n^2 + 4n + 7 \le 14n^2$

$14 = 14$

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

$(n \log n)$

Constant time

$O(1)$

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

1 1 1 1 1

$1 + 1 + 1 + 1 + 1$

$T = (n+1)$

$O(n)$

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

$1 + \underbrace{1 \quad 1 \quad \ldots \quad 1}_{n-1} + 1$

$1 + n-1 + 1$

$T(n) = n+1$

$O(n)$

```
public static int Max1(int[] arr) {
        int max = arr[0];
        for (int i = 1; i < arr.length; i++) {
                if (arr[i] > max) {
                        max = arr[i];
                }
        }
        return max;
}
```

```
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;
}
```

$O(\log n)$

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

$N = 2^k$

$2^k = n$

$k = \log_2 n$

$O(k)$

$\to 0 \quad \frac{n}{2^0} \to 1$

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

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

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

$\frac{n}{2^k} \quad 1 \to 1$

k

i=1 i=2 $i=2^2$ $i=2^3$ $i=2^4$ --- $i=2^k \le n$

1 2 1 . 1

$2^k = n$

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

```
for(i= 1; i<= n; i++) {
        for(int j= 1; j<= n; j++) {
                // O(N^2)
                System.out.println("hey");
        }
}
```

$j = 1 \quad n$

$j = 2 \quad n$

$j = n \quad n$

i=1 i=2 i=3 i=4 i=5 --- i=n

N n n n n n

$n \times n$

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

N^4

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

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

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

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

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

$\frac{1}{n} \to \log(n)$

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

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

$N\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{N}\right)$ $\log(n)$

$n \times \log(n)$

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

n n-1 n-2 1

$1 + 2 + 3 \ldots n$

$\frac{n(n+1)}{2} \quad n^2$