CS & IT ENGINEERING

Algorithms

Lecture No.- 01

1500 Series

By- Dr. Khaleel Khan sir

Recap of Previous Lecture

Topic

Topics to be Covered

Topic Analysis of Algorithms

Topic Divide and Conquer

Math Background.

- Logs
- Exponents
- Sum Series

1) Analysis

- Time Complexities
  - ASN's
  - Recursive & Non-Recursive Alg's
  - Loop Complexities

2) Divide & Conquer

3) Greedy Method

4) Dynamic Programming

5) Graph Techniques

6) Heap Algorithms

7) Sorting Tech's

#Q. Given three algorithms for solving a graph problem having n vertices & e edges. Algorithm 'A' takes n^2 ms, algorithm B takes 100n ms; while C takes $100\left(2^{n/10}-1\right)ms$.

Which algorithm is fastest for all very large value of n?

$A \rightarrow n^2$ ms
$B \rightarrow 100 \times n$ ms
$\}$ Poly

$C \rightarrow 100\left(2^{n/10}-1\right)$ ms $\}$ Expo.

[Poly < Expo]

$\boxed{B < A < C}$

Fastest

Slowest

A. A

B. B

C. C

D. All are equally fast.

#Q. Let B(n) and A(n) respectively denote the best case and average case running time of an algorithm of input size 'n' which of the following is always false?

A. $B(n) = \Omega(A(n))$ May be True

B. $B(n) = \theta(\cancel{W}A(n))$ May be TRUE

C. $B(n) = w(A(n))$

small omega

D. $B(n) = O(\cancel{W}A(n))$ May be TRUE

$$\boxed{B(n) \le A(n) \le W(n)}$$

$\boxed{B(n) > A(n)}$ Always false

#Q. $f(n) = \sum_{i=1}^{n} i^{\frac{1}{2}}$ is____.

A $\theta(n)$

B θ (nlogn)

C $\theta\left(n\sqrt{n}\right)$

D $\theta\left(\sqrt{n}\right)$

$$f(n) = \sum_{i=1}^{n} i^{1/2} \sim \int_{1}^{n} i^{1/2} = \left[\frac{i^{1/2+1}}{\frac{1}{2}+1}\right]_{1}^{n}$$

$$\int x^{n} = \frac{x^{n+1}}{n+1}$$

$$= \left[\frac{n^{3/2}}{} \times \frac{2}{3}\right]$$

$$= \Theta\left(n^{1.5}\right)$$

$$= \Theta\left(n\sqrt{n}\right)$$

$$f(n) = \sum_{i=1}^{n} i^{-1/2} = O(\quad)$$

$$= \int_{1}^{n} 1/i^{1/2} = \log n^{1.5} \Rightarrow 1.5 \times \log n$$

$$= O(\log n)$$

#Q. Let k be an integer greater than 1. Which of the following represents the order of growth of the expression $\sum_{i=1}^{n} k^i$ as a function of n ?

$$S_n = \frac{a(r^n - 1)}{r-1}$$

$$= \frac{\cancel{k}(k^n - 1)}{\cancel{k-1}} = \Theta(k^n)$$

A $\Theta(k^n)$

B $\Theta(k^{n \log n})$

C $\Theta(k^n \text{ lon } n)$

D $\Theta(k^{2kn})$

E $\Theta(n^{k+1})$

#Q. Consider the following functions:

$$f(n) = n^2 \log n$$
$$g(n) = n \log^{10} n$$
$$h(n) = n^3$$

Inc. order : $n (\log n)^{10} < n^2 \cdot \log n < n^3$

$g < f < h$

which of the following is incorrect?

A. $g(n) = O(f(n))$: T

B. $g(n) = O(h(n))$: T

C. $f(n) = O(h(n))$: T

D. None of the above

#Q. Consider the functions below:

$\log_2(n!), \frac{n}{2^n}, \frac{n}{\log_2 n}, (\log_2 n)!, \log_2 n$

$(n\log n)$

(1) (2) (3) (4) (5)

P P P E P

Arrange them in the decreasing order of rates of growth.

$x! = O(x^x)$

$(\log n)! = (\log n)^{\log n}$

a) 1, 4, 5, 3, 2

b) 1, 4, 3, 5, 2

c) 2, 3, 5, 4, 1

d) 2, 3, 5, 1, 4

e) 4, 1, 5, 3, 2

$(\log n)! > \log(n!) >$

$(4) > (1) > (3) > (5) > (2)$

$\boxed{\frac{n}{2^n} \quad \frac{n}{\log n}}$

(2) (3)

$\log n$

(5)

$\frac{n}{\log n} > \log n > \frac{n}{2^n}$

#Q. Arrange the following functions in the increasing order of their rates of growth:

E E P P P E E E E

$(\text{sqrt}(2))^n$, $2^{\text{sqrt}(n)}$, $n^2\log n$, $n(\log n)^2$, $(n\log n)^2$, $n^{\log n}$, $n^{\text{sqrt}(n)}$, n^n, $(\log n)^n$.

(1) (2) (3) (4) (5) (6) (7) (8) (9)

$\langle 2 \quad 1 \rangle$

$(\sqrt{2})^n > 2^{\sqrt{n}}$

$n(\log n)^2 < n^2 \log n < n^2 \log^2 n$

$(4 \quad 3 \quad 5)$

6	7	8	9
$n^{\log n}$	$n^{\sqrt{n}}$	n^n	$(\log n)^n$

6 7 9 8

$(4 \; 3 \; 5 \; 6 \; 2 \; 1 \; 7 \; 9 \; 8)$

$n^{\sqrt{n}} < (\log n)^n$

$\cancel{\sqrt{n}} \cdot \log n \quad \cancel{\sqrt{n}} \cdot \sqrt{n} \log \log n$

a) 4 3 5 6 7 9 1 8

b) 4 3 6 9 1 2 5 7 8

c) 6 5 4 3 2 1 7 8

d) 4 5 3 1 2 9 6 8

#Q. Consider the equality

MSQ

$$\left[\sum_{i=0}^{i=n+1} 2^i\right]\left[\sum_{i=0}^{i=n} = \frac{1}{2^i}\right] = P$$

$$\frac{1\left(2^{n+2}-1\right)}{2-1} \qquad \frac{1\left(1-\frac{1}{2^{n+1}}\right)}{1-\frac{1}{2}}$$

<A+B+C>

Which of the following notation asymptotically is / are remains correct, with respect to P?

$$\left(2^{n+2}-1\right)\left(1-\frac{1}{2^{n+1}}\right)$$

$$= 2^{n+2} - 2 - 1 + \frac{1}{2^{n+1}}$$

$$= \left[2^{n+2} - 3 + \frac{1}{2^{n+1}}\right]$$

A $\theta(2^{n+2})$ ✓

B $\Omega(2^n)$ ✓

C $O(2^{2n})$ ✓

D $\theta(2^{2n})$ ✗

$(2^2)^n = 4^n$

#Q. Consider the following C function:

```
main()
{
    for (i = 1; i ≤ n; ++i)
    {
        j = 1;
        while ( j ≤ n)
            j = 2*j;
        for (k = 1; k ≤ n; ++k)
            c = c + 1;
    }
}
```

The time complexity of above algorithm will be?

A. $O(n \log n)$

B. $O(n^2 \log n)$

C. $O(n^2)$

D. None of the above

for loop: n — Total

1. $j = 1$: n

2. while loop: $\log n$: $n \log n$

3. for loop: n : n^2

$n + n \cdot \log n + n^2 = O(n^2)$ ✓

The return (Final) value of K is $\Theta(\_\_\_\_)$;

#Q. Consider the following function:

```
int unknown (int n)
{
    int i, j, k = 0;
    for (i = n/2; i < n; i++)
    {
    for (j = 2; j ≤ n; j = j * 2)
        k = k+ n/2;
    }
    return (k);
}
```

$K = \left(\frac{n}{2} * \log n\right) * \frac{n}{2}$

$\Theta(n^2 \log n)$

What will be the time complexity of above C code?

A. $\theta(n^2 \log n)$

B. $\theta(n \log n)$

C. $\theta(n^2)$

D. None of the above

#Q. Consider the following code segment

$n * 3 * \log n$

$= O(n \log n)$

```
for (int x = 0; x < n; x ++)  : n
for (int y = 1; y < = n*n*n; y = 2*y)
    print("spread positivity");
```

$n*n*n = z$

$\Longrightarrow \log z = \log n^3 = 3 * \log n$

What is the running time complexity of the above code?

A $\theta(n^2)$

B $\theta(n \log n)$

C $\theta(n(\log n)^3)$

D $\theta(n \log \log n)$

#Q. If the given program takes 10 milliseconds to run n = 50, how long, in milliseconds, will it take to run for n = 500? ____.

```
addn = 0;
for (a = 0; a < n/2; a++)
for (b = 0; b < n/4; b++)
    addn ++;
```

$O(n^2)$

(1000 ms)

$(50)^2$ units $\Longrightarrow$ 10 ms

1 unit $\Longrightarrow$?

$$\boxed{1 \text{ unit} = \frac{1\cancel{0}}{50 \times 5\cancel{0}} = \frac{1}{250} \text{ ms}}$$

$(500)^2$ units $\Longrightarrow$?

$$\frac{\cancel{5}00 \times \cancel{5}0\cancel{0}}{\cancel{2}5\cancel{0}} = \underline{1000 \text{ ms}}$$

#Q. Determine the Time Complexity of the following iterative function:

```
int f (int A[SIZE] [SIZE], int n)
{
    int i, j, sum = 0;
    for (i = 1; i ≤ n; ++i)
    {
        if (i % 2 ==0)
            for (j =1; j ≤ i; j = j + 1) sum = sum + A[i] [j];
        else
            for (j = n ; j > i; j = j – 1) sum = sum – A[i] [j];
    }
}
```

$i=1 \quad i=2 \quad i=3 \quad i=4 \quad i=5 \quad i=6$

Total Step Count $= (n-1) + 2 + (n-3) + 4 + (n-5) + 6 \cdots + 1 + n$

$= 2+4+6+\cdots+n + n*\frac{n}{2} - \left(1+3+5+\cdots+\underline{n-1}\right)$

$= \frac{\frac{n}{2}\left(\frac{n}{2}+1\right)}{2} + \frac{n^2}{2} + \cdots = O\left(n^2\right)$

#Q. What does the following function compute ?

```
int F (int n, int *Fprev)
{
    int Fn_1, Fn_2;
    if (n == 0) {
        *Fprev = 1;
        return (0);
    }
    if (n==1) {
        *Fprev = 0;
        return (1);
    }
    Fn_1 = F(n -1, & Fn_2);
    *Fprev = Fn_1;
    return (Fn_1+Fn_2);
}
```

Space: $O(n)$

Recursion Stack

Return n^{th} Fibonacii Number

Time: $O(n)$ ✓

0 1 1 2 3 5 . . .

n: 0 1 2 3 4

Time Complexity Recurrence of

F(n)

$$T(n) = c, \quad n \le 1$$
$$= a + T(n-1) + b$$
$$= T(n-1) + d, \quad n > 1$$

2 mins Summary

Topic	One	Analysis of Algorithms
Topic	Two	Divide and Conquer
Topic	Three	
Topic	Four	
Topic	Five	

THANK - YOU