i. $4n+100=O(n)$
ii. $500n^3+6n+6=O(n^3)$
iii. $n^3 \neq O(n^2)$
iv. $5n^2-6n= O(n^2)$
v. $n! = O(n^n)$
vi. $2n^2 2^n +n\log n= O(n^2 2^n)$
vii. $3n^3 +4n^2 = \Omega(n^2)$
viii. $\sum_{i=0}^{n} i^2=\theta(n^3)$

$\log x^y = y \log x$

$\log x \cdot y = \log x + \log y$

$\log_a a = 1$

3. Compare the two functions n^2 and $2^n/4$ for various values of n. Determine when the second becomes larger than the first.

$n = 2^{10}, 2^{1024}$ 2^{2048}

4. Arrange the following in the increasing order of asymptotic complexity of functions f1, f2, f3 and f4.
f1(n) = 2^n, f2(n) = n^(3/2), f3(n) = nLogn, f4(n) = n^(Logn)

$f_1 = 2^n$, $f_2 = n^{3/2}$, $f_3 = n\log n$, $f_4 = n^{\log n}$ 2^{1024}

$\log_2 2^n$, $\log n^{3/2}$, $\log(n\log n)$, $\log n^{\log n}$

n, $3/2 \log n$, $\log n + \log\log n$, $\log^2 n$

2^{10}, 15, 13, 100 $(1024)^2 = 2^{20}$

2^{1024}, 1536, 1034,

$f_3 < f_2 < f_4 < f_1$

$f_3 < f_2 < f_4 < f_1$

Jaypee University of Engineering and Technology

18B11CI311 – Data Structures
B.Tech -3rd Semester
Tutorial – 2 (Time Complexity)

Q1. Analyze the time complexity for following code segments/ functions.

i.
```
1. int count = 0;
2. for (int i = n; i > 0; i /= 2)
3.         for (int j = 0; j < i; j++)
4.                 count += 1;
```

ii.
```
for (i=1; i<=n; i++)
    for (j=1; j<=log(i); j++)
        PRINT "HELLO JUET !" ;
```

iii.
```
if (n==1)
   return;
for (int i=1; i<=n; i++)
{
     for (int j=1; j<=n; j++)
     {
          printf("*");
          break;
     }
}
```

iv.
```
void fun(int n, int arr[])
1. {
2.     int i = 0
3.     for(; i < n; i++)
4.       for(j=0;j < n && arr[i] < arr[j];j++)
5.           Print"i+j";
6. }
```

v.

vi.

i j
1 log(1)
2 log(2)
3 log(3)
⋮ ⋮
log(n) n

$O(n \log n)$

$\log(n!) \approx n \log n$

$f(n) = \log(1) + \log(2) + \log(3) + \ldots + \log n$
$= \log(1 \cdot 2 \cdot 3 \cdot \ldots \cdot n)$

B.Tech -3rd Semester

Tutorial – 3

(Arrays)

1. Consider an array of 20 elements is stored in the memory of 40 bytes from 200 to 238. Assume array index starts from 3. Find out the address of 10th index element.

2. Consider an array of 30 elements is stored in the memory of 120 bytes from 3000 to 3116. Find out the address of 13th index element.

3. Consider a 2D array A of 30 (5x6) elements is stored in the memory of 120 bytes from 2100 to 2216. Find out the address of A[2][4] element in row-major and column-major order.

4. In a 2D integer array TD, assume that the row indices range from -3 to 7 and column indices range from 6 to 14. An element TD [-3, 6] stored at address 3220. Find out the dimension of TD and address of an element TD [2, 10], if TD stores the elements in column major order.

5. Assume you have given an array A[15][20]. Each element needs 'W' bytes of storage. If the address of A[6][8] is 4440 and the base address at A[1][1] is 4000, find the width 'W' of each cell in the array A when the array is stored as Column Major Wise.

6. Consider a 2D array A[m][m], each element takes 4 bytes of storage. If the base address at A[1][1] is 1500 and the address of A[4][5] is 1608, determine the order of the matrix when it is stored in Column Major Wise.

W = 2 B

base(a) = 200

LB = 3

i = 10

Loc (A[i]) = base(a) + w(i - LB)

= 200 + 2 (10 - 3)

= 214