DAA-QUIZ-2

Q1. Do the time analysis in the aspect of Theta (θ) of the below-given pseudocode.

```
1.  $for\ i \leftarrow 1\ to\ n$
2.  $do$
3.          $for\ j \leftarrow 1\ to\ 2i$
4.          $do$
5.                  $k = j$
6.                  $while\ (k \geq 0)$
7.                  $do$
8.                          $k = k - 1$
9.                  $end\ while$
10.         $end\ for$
11. $end\ for$
```

Ans:

Lets first solve inner loop i.e. from line no. 4 to 6

As we know the while loop based on the value of k where $k = j$

Lets say $T(J)$ for while loop

$$J = \sum_{k=0}^{j} 1 = j + 1$$

Hence $T(J) = j + 1$

Now calculate the inner for loop i.e. from 3 to 6

Lets say

$$T(I) = \sum_{j=1}^{2i} T(J) = \sum_{j=1}^{2i} (j + 1)$$

$$= \sum_{j=1}^{2i} j + \sum_{j=1}^{2i} 1$$

$$= \sum_{j=1}^{2i} j + 2i - - - - - - - - - eq1$$

Now calculate summation of j by applying summation rule arithmetic series we get:

$$\sum_{j=1}^{2i} j = \frac{n(n+1)}{2}\ in\ our\ case\ n = 2i\ so\ equation\ would\ be$$

$$= \frac{2i(2i+1)}{2}\ please\ this\ value\ in\ the\ eq1\ we\ get$$

$$T(I) = \frac{2i(2i+1)}{2} + 2i$$

Solve the above equation we get

$$T(I) = \frac{\cancel{2}i(2i+1)}{\cancel{2}} + 2i = 2i^2 + i + 2i = 2i^2 + 3i$$

Hence running time of inner loop is

$$T(I) = 2i^2 + 3i$$

Now calculate the outer-most loop i.e. from line no. 1 to so on.

$$T(n) = \sum_{i=1}^{n} T(I) = \sum_{i=1}^{n} (2i^2 + 3i)$$

$$T(n) = 2\sum_{i=1}^{n} i^2 + 3\sum_{i=1}^{n} i$$

$$T(n) = 2\sum_{i=1}^{n} i^2 + 3\frac{n(n+1)}{2}$$

$$T(n) = 2\frac{2n^3 + 3n^2 + n}{6} + 3\frac{n(n+1)}{2}$$

$$T(n) = \frac{4n^3 + 6n^2 + 2n}{6} + 3\frac{n(n+1)}{2} = \frac{4n^3 + 6n^2 + 2n}{6} + 3\frac{n^2 + n}{2}$$

$$T(n) = \frac{4n^3 + 6n^2 + 2n}{6} + \frac{3n^2 + 3n}{2}$$

Remember the LCM in above case it would be 6

So

$$T(n) = \frac{4n^3 + 6n^2 + 2n}{6} + \frac{9n^2 + 9n}{6} = \frac{4n^3 + 6n^2 + 2n + 9n^2 + 9n}{6}$$

$$T(n) = \frac{4n^3 + 15n^2 + 11n}{6}$$

Hence the highest value of n is n^3

Thus $\boldsymbol{\theta(n^3)}$

Q2. The Merge sort is a sorting technique based on the divide and conquer technique; the Merge sort divides the array into equal halves and then combines them in a sorted manner.

a. Write a pseudocode of merge sort
b. Write a recurrence relation of merge sort and expand terms up to the n size of an array, where $n = 8$.

Ans: a.

```
$\boldsymbol{MERGE\_SORT(array\ A, int\ p, int\ r)}$
$if\ (p < r)$
$then$
    $q \leftarrow (p + r)/2$
    $MERGE\_SORT(A, p, q)$
    $MERGE\_SORT(A, q + 1, r)$
    $MERGE(A, p, q, r)$
$end\ if$
```

```
$\boldsymbol{MERGE(array\ A, int\ p, int\ q, int\ r)}$
$int\ B[p \ldots r];$
$int\ i \leftarrow k \leftarrow p;$
$int\ j \leftarrow q + 1;$
$while(i \leq q)\ and\ (j \leq r)$
    $do\ if(A[i] \leq A[j])$
        $then\ B[k++] \leftarrow A[i++]$
        $else\ B[k++] \leftarrow A[j++]$
$end\ while$
$while(i \leq q)$
    $do\ B[k++] \leftarrow A[i++]$
$end\ while$
$while(j \leq r)$
    $do\ B[k++] \leftarrow A[j++]$
$end\ while$
$for\ i \leftarrow p\ to\ r$
$do\ A[i] \leftarrow B[i]$
$end\ for$
```

b.

$T(1) = 1$

$T(2) = T(1) + T(1) + 2 = 1 + 1 + 2 = 4$

…

$T(4) = T(2) + T(2) + 4 = 4 + 4 + 4 \ = 12$

…

$T(8) = T(4) + T(4) + 8 = 12 + 12 + 8 \ = 32$