Tutorial Sheet – EVEN 2021

15B11CI411 – Algorithms and Problem Solving

Instructions

1. Tutorials in APS course would majorly be doubt removal sessions.
2. Students are advised to come prepared in tutorial by revising the lectures of that particular week so that doubts related to topics covered in that week are resolved.
3. Few practice questions will also be posted as Tutorial Sheet on your Google Classrooms. You can try solving them before the scheduled tutorial and discuss the doubts.
4. As tutorials will be purely problem solving based, always join the tutorial with a notebook and pen with you.
5. Whatever questions are discussed in tutorial, you are supposed to submit them on your respective classrooms every week.

Week 2 (25th January to 30th January 2021)
Asymptotic Analysis and Notations

Q1. Analyze the time complexity of following functions:

(a) **Function: One** ()	(b) **Function: Two** (int n)	(c) **Function: Three** (int n)
| ```
{
    int x; int i; int n;
    x = 20;
    input n;
    for(i = 0; i < n; i++)
        x++;
    output x;
}
``` | ```
{
    int *x; int i;
    allocate memory for x to
        store n elements
    for(i = 0; i < n; i++)
    {
        input x[i];
        x[i] = x[i] + i;
        output x[i];
    }
}
``` | ```
{
    int *x; int i;
    allocate memory for x to
        store n elements
    for(i = 0; i < n; i++)
        input x[i];
    for(i = 0; i < n; i++)
        x[i] = x[i] + i;
    for(i = 0; i < n; i++)
        output x[i];
}
``` |
| (d) **Function: Four** (int n, int y) | (e) **Function: Five** (int n) | (f) **Function: Six** (int m, int n) |
| ```
{
    int *x; int i; int j;
        j = 0;
    allocate memory for x to
        store n elements
    for(i = 0; i < n; i++)
            input x[i];
    for(i = 0; i < n; i++)
    {
            if(x[i] == y)
            j++;
    }
    if(j > 0)
        output y is present j times
    else
        output y is not present
}
``` | ```
{
    int *x; int i; int j, int m; int t;
    allocate memory for x to
        store n elements
    for(i = 0; i < n; i++)
            input x[i];
    for(i = 0; i < n; i++)
    {
        m = x[i];
        t = i;
        for(j = i+1; j < n; j++)
        {
            if(m > x[j])
            {
                m = x[j];
                t = j;
            }
        }
        x[t] = x[i];
        x[i] = m;
    }
    for(i = 0; i < n; i++)
        output x[i];
}
``` | ```
{
    int **x; int i; int j; int s;
    allocate memory for x to
        store m*n elements
    s = 0;
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
            input x[i][j];
    }
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
            s = s + input x[i][j];
    }
    output s;
}
```
//**formulate** the required algorithmic time for above function when (a) m = n (b) m ≠ n |

Q2. Analyze the time complexities of following in terms of Big O and Big Omega:

(a) $F(n) = 3\,n^2 + 2\,n + 6$
(b) $F(n) = n^4 + 6\,n + 3$
(c) $F(n) = 2\,n^3 + 3\,n^2 + n + 4$

Q3. Decide whether these statements are True or False. You must briefly justify all your answers.

1. If $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$, then $h(n) = \Theta(f(n))$

2. If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $h(n) = \Omega(f(n))$

3. If $f(n) = O(g(n))$ and $g(n) = O(f(n))$ then $f(n) = g(n)$

4. $\dfrac{n}{100} = \Omega(n)$

Q4.
Suppose $T_1(N) = O(f(N))$ and $T_2(N) = O(f(N))$. Which of the following are true?

(a) $T_1(N) + T_2(N) = O(f(N))$
(b) $\frac{T_1(N)}{T_2(N)} = O(1)$
(c) $T_1(N) = O(T_2(N))$

Q5. If T(n) is $O(n^{4/3})$ which of the following is False?

a) T(n) is O(n log n)
b) T(n) is O(n²)
c) T(n) is O(n²log n)
d) T(n) is O(n³)