BIM304 - Computer Algorithm Design
Assignment I

Please show all your work! Answers without supporting work will not be given credit. You have to write answers in spaces provided. Please upload your paper until **21.03.2022** to the **MERGEN**.

Name: *ID:*

1-) Write functions (in Java) that draw Pyramid with "*" given height and stars variables. Make implementations for both **iterative** and **recursive** design. **(20 P)**

For Example: Both methods will draw a pyramid according to the height value entered on the screen.

int height = 5; // can be any number of type integer

int stars = 1; //**this variable will be entered as 1 by default when calling the method**

IterativePrintPyramid(int height) **(10p)**

```
    *
   ***
  *****
 *******
*********
```

RecursivePrintPyramid(int height, int stars) **(10p)**

```
    *
   ***
  *****
 *******
*********
```

```
void IterativePrintPyramid(int height){

    } // end−of−IterativePrintPyramid
```

```
void RecursivePrintPyramid(int height, int stars) {

    } // end−of−RecursivePrintPyramid
```

2-) Prove or disprove the given equivalence. **(20 P)**

a-) $f(n) = \frac{n^2}{150} = \Omega(n)$ **(5p)**

b-) $f(n) = 5n^2 + 8n + 15 = O(n^2)$ **(5p)**

c-) $f(n) = 7n^2 - 3n + 2 = O(n)$ **(5p)**

d-) $f(n) = 5n^3 + 2n^2 + 4n + 6 = \theta(n^3)$ **(5p)**

3-) Express the running time of the following recurrence in Big-O by solving the recurrence using repeated expansion. **(10 P)**

$$T(n) = \begin{cases} 1 & n = 1 \\ T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + 1 & n > 1 \end{cases}$$

4-) For each of the following code segments, give an analysis of the running time (Big-O) and find the number of “*” will be printed in terms of n? **(20 P)**

(1)

```
for(i=0; i<n; i++)
   for(j=0; j<n*n; j++)
      print(“*”);
```

(2)

```
for(i=0; i<n; i++)
   for(j=0; j<i; j++)
      print(“*”);
```

(3)

```
for(i=0; i<n; i++)
   for(j=0; j<i; j++)
      for(k=0; k<j; k++)
         print(“*”);
```

(4)

```
for(i=0; i<n; i++)
   for(j=0; j<n; j++)
      for(k=0; k<j; k++)
         print(“*”);
```

Code Fragment	Big-O **(Each of 2 points)**	Number of “*” asterisk printed in terms of n **(Each of 3 points)**
1		
2		
3		
4		

5-) Sort the given array in ascending order using the **Bubble Sort**, **Insertion Sort** and **Selection Sort** algorithms. Show the position of the elements in the array at **each iteration**. **(30 P)**

25	80	52	36	1

Bubble Sort **(10p)**

Insertion Sort **(10p)**

Selection Sort **(10p)**