DAA Assignment I

1. **Let A be the adjacency matrix of an undirected graph. Defining each of the following, explain what property of the matrix indicates that**

    i. **the graph is complete.**

    ii. **the graph has a loop, i.e., an edge connecting a vertex to itself.**

    iii. **the graph has an isolated vertex, i.e., a vertex with no edges incident to it.**

A graph is a mathematical structure consisting of a set of vertices and a set of edges connecting the vertices.

i. A complete graph has all of its vertices connected to every other vertex which may or may not include self loops to that vertex.

   The matrix is filled with ones in all the places (which may not include the primary diagonal).

ii. A graph with a loop has a connection from a vertex to itself.

   Any one of the element in the primary diagonal is non zero, i.e. '1'.

iii. A graph has an isolated vertex when the vertex has no connection to any of the other vertices in the graph.

   One row and column is filled with zeros.

2. **Let A be the adjacency list of an undirected graph. Defining each of the following, explain what property of the list indicates that**

    i. **the graph is complete.**

    ii. **the graph has a loop, i.e., an edge connecting a vertex to itself.**

    iii. **the graph has an isolated vertex, i.e., a vertex with no edges incident to it.**

A graph is a mathematical structure consisting of a set of vertices and a set of edges connecting the vertices.

i. A complete graph has all of its vertices connected to every other vertex which may or may not include self loops to that vertex.

    Every element in the list's header has *n - 1* or *n* links.

    That is *'n - 1'* or *'n'* connections from one vertex to all of the other vertices (may to may not including the self loop.)

    (Given that there are 'n' vertices in the graph.)

ii. A graph with a loop has a connection from a vertex to itself.

    A list's header has a link with value equal to the header's value.

iii. A graph has an isolated vertex when the vertex has no connection to any of the other vertices in the graph.

    One header of the list has no links. (No connections to any other vertex.)

3. **Arrange the following functions in increasing order of their order of growth (from lowest to highest) After arrangement prove that the functions are listed in increasing order of their order of growth.**

log n, n, n log n, n^2, n^3, 2^n, $n!$

[1 + 3]

$\log(n) < n < n\log(n) < n^2 < n^3 < 2^n < n!$

If

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty,$$

then the growth rate of $f(n)$ is larger than the growth rate of $g(n)$. Vice versa, if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0,$$

then the growth rate of $f(n)$ is smaller than the growth rate of $g(n)$.

$$\lim_{n\to\infty} \frac{n}{\log(n)} = \infty \qquad \lim_{n\to\infty} \frac{n\log(n)}{n} = \infty$$

$$\lim_{n\to\infty} \frac{n^2}{n\log(n)} = \infty \qquad \lim_{n\to\infty} \frac{n^3}{n^2} = \infty$$

$$\lim_{n\to\infty} \frac{2^n}{n^3} = \infty \qquad \lim_{n\to\infty} \frac{n!}{2^n} = \infty$$

(Prove it for the glory of Satan if you must)

4. A. Consider the following algorithm.

```
ALGORITHM Enigma(A[0..n − 1, 0..n − 1])
// Input: A matrix A[0..n − 1, 0..n − 1] of real numbers
    for i ←0 to n − 2 do
        for j ←i + 1 to n − 1 do
            if A[i, j ]≠ A[j, i]
                return false
    return true
```

i. What does this algorithm compute? (½)
ii. What is the efficiency class of this algorithm (1½)

i. This algorithm checks if the inputted square matrix is symmetric or not.

ii. Efficiency: $O(n(n-1)/2) = O(n^2) = \Theta(n^2)$

$$\sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (-i+n-1) = \frac{1}{2}(n-1)n$$

4. B.

i. Design a recursive algorithm for computing 2^n for any nonnegative integer n that is based on the formula $2^n = 2^{n-1} + 2^{n-1}$. (1)
ii. **Set up a recurrence relation for the number of additions made by the algorithm and solve it. (1)**

```
i. // Algorithm PowTwo (n)
    // Input: An non negative integer n
    // Output: 2^n

    If n = 0
        return 1
    else
        return PowTwo(n−1) + PowTwo(n−1)
```

ii. For $n = 0$, Number of additions = 0;

For $n = 1$, number of additions = 1;

For $n = 2$, number of additions = $1 + (1 + 1) = 3$;

For $n = 3$, number of additions = $1 + ((1 + (1 + 1) + (1 + (1 + 1)) = 7$;

For n elements, number of additions =

$$
\begin{aligned}
&= 1 + 2 * (1 + (2 * (1 + \ldots))) \\
&= 1 + 2^2 + 2^3 + \ldots + 2^{n-1} \\
&= (2^n - 1)/(2 - 1) \\
&= 2^n - 1
\end{aligned}
$$

5A. Is selection sort stable? Justify your answer with the help of an example. (½ + 2)

Selection sort is not a stable sorting algorithm.

Selection sort algorithm picks the minimum and swaps it with the element at current position.

For e.g.

Suppose the array is : 5 2 3 8 4 5 6

Let's distinguish the two 5's as 5_a and 5_b

So our array is: 5_a 3 4 5_b 2 6 8

After iteration 1: 2 will be swapped with the element in 1st position:

So our array becomes: 2 3 4 5_b 5_a 6 8

Since now our array is in sorted order and we clearly see that 5_a comes before 5_b in initial array but not in the sorted array.

5B. Find the average number of key comparisons required for sequential search. (1½)

The fewest number of comparisons = 1

The maximum number of comparisons = N, (N being the size of the list)'

The number of comparisons for i^{th} item on the list = i.

$$\therefore \text{The average number of comparisons} = \frac{1 + 2 + 3 + \ldots + N}{N} = (N + 1) / 2$$