Homework 05:

Due date: Wednesday, 08/01 at 11:59 pm

Instructions:

Submit a typed or neatly handwritten scan of your responses on Canvas in PDF format.

Note: you will need to submit a separate PDF per each section.

1. Pseudocode to Merge Two Sorted Arrays

Write a pseudocode for the Merge procedure that is used in Merge Sort. In other words, given two sorted arrays *left* and *right* (each containing comparable elements like integers or strings) as input, the method must output a sorted array that contains all elements of *left* and *right*. The following properties must be satisfied by your algorithm:

(a) If *left* is of length n and *right* is of length m, then the runtime of the method must be $O(n + m)$.

(b) The input arrays *left* and *right* should remain unmodified.

(c) The algorithm must be *stable*. In other words, if elements of the same rank are present in both *left* and *right*, then the output array must have the element from *left* preceding the element from *right*. If multiple elements of the same array are of equal rank, they must appear in the output in the same order that they are in the input.

2. Modeling as Graphs

Alice is throwing a party and invites her friends Bob, Claire, and Dan. Bob is only friends with Alice. Dan and Claire are friends. Dan brings along his friend Eve to the party, and Eve is not friends with Alice, Bob, or Claire.

(a) Model the friendship relationships as a graph. Party-goers are vertices, and edges are friendships. For simplicity, you can label the vertices by A,B,C,D, and E, in which A denotes Alice, B denotes Bob and so on.

(b) Write down the adjacency matrix of the above graph. (use $A \rightarrow 0, B \rightarrow 1$, etc.)

(c) Write down the adjacency list of the above graph. (use $A \rightarrow 0, B \rightarrow 1$, etc.)

(d) Which vertex has the highest degree (give the person's name)? Which has the lowest? What are the degrees of these vertices? What does vertex degree tell you about the person that vertex represents?

1. First, we assum there're 2 arrays called "Left" and "Right"
We set a method, "sort", to make the final array what includes these two arrays.

```
    T[] Sort (T[] Left, T[] right)
```

Second, we set a new array T[left.length + right.length] as the final sorted array. We set index of "left" (i), index of "right" (j), index of the new array (final one: k), and two integers m (left's length) & n (right's length)

Then, we do the "while" loop based on the conditions.

```
    while (i+j < n+m && i<n && j<m)
        if (left[i] ≤ right[j])
      Then the final array (new array) [k++] = left[i]
           i ++
        else, the new array [k++] = right [j]
```

However, if the left array are sorted before the right array. (i=n), the left is empty. then add the right array.

```
      for (int o = j; o<m; o++), the "o" is the next index
            new array [k++] = right [o]   of "j"

  if the right array is empty first, (j=m)
      for (int o = i; o<n; o++)
            new array [k++] = left [o]
```

Last, we return the new array as our final array.