Lab2 - Answers

Saurab Ghimire
417555

Problem 1: Solution
The asymptotic running time of the given procedure is $O(n^2)$. The first for loop is O(n). The second for loop has a inner for loop so it is O(n.n). The O notation asymptotic running time of the procedure therefore is $O(n^2)$.

Problem 2: Solution
The pseudo code to merge two sorted arrays together is as follows:

```
merge(arr1, arr2):
    initialize new list : length of arr1 + length of arr2
    let left = 0, right = 0

    while left < length(arr1) AND right < length(arr2):
        if arr1[left] < arr2[right]:
            arr1[left] to mergedList
            left increment
        else:
            arr2[right] to mergedList
            right increment

    append remaining elements of arr1 to mergedList
    append remaining elements of arr2 to mergedList

    return mergedList
```

The asymptotic running time is O(n+m).

Implemented code:

```java
public class MergeAlgorithm {

    public static void main(String[] args) {
        int[] arr1 = {1, 4, 5, 8, 17};
        int[] arr2 = {2, 4, 8, 11, 13, 21, 23, 25};
        int[] mergedList = merge(arr1, arr2);
        System.out.println(Arrays.toString(mergedList));
    }

    1 usage
    public static int[] merge(int[] arr1, int[] arr2) {
        int[] result = new int[arr1.length + arr2.length];
        int left = 0, right = 0, index = 0;

        while (left < arr1.length && right < arr2.length) {
            if (arr1[left] < arr2[right]) {
                result[index] = arr1[left];
                left++;
            } else {
                result[index] = arr2[right];
                right++;
            }
            index++;
        }

        while (left < arr1.length) {
            result[index] = arr1[left];
            left++;
            index++;
        }

        while (right < arr2.length) {
            result[index] = arr2[right];
            right++;
            index++;
        }

        return result;
    }

}
```

Problem 3: