Timothy Hoang

4/24/19

CMPT435-111

Assignment 10

**Date Assigned: 04/15/2019**

**Due: Midnight 04/24/2019 on iLearn**

**Please read turn-in checklist at the end of this document before you start doing exercises.**

**Section 1: Pen-and-paper Exercises**

1. Trace Merge Sort on the following array:

{1, 7, 6, 8, 0, 2, 5}

**Note: Write down the state of the array after each recursive call. (5 points).**

1, 7, 6, 8 0, 2, 5

1,7 6,8 0,2 5

1 7 6 8 0 2 5

1,7 6,8 0,2 5

1,6,7,8 0,2,5

{0,1,2,5,6,7,8}

1. Below is the pseudocode for Quicksort and Partition that we talked about in class. As usual with recursive functions on arrays, we see the array indices s and e as arguments. Quicksort(A, s, e) sorts the part of the array between s and e inclusively. The initial call (that is, to sort the entire array) is Quicksort(A, 0, n − 1).

QuickSort(A, s, e)

if s < e

p = Partition (A, s, e) // Partition the array and return the position of pivot after the partition

QuickSort(A, s, p-1) // Sort left side

QuickSort (A, p+1, e) // Sort right side

end if

Partition(A, s, e)

pivot = A[s], i = s + 1, j = e; // Let the leftmost element be the pivot

while i<=j // Rearrange elements

while i < e & A[i] < pivot,

i = i + 1

end while

while j > s & A[j] >= pivot,

j = j - 1

end while

if i >= j

break

end if

swap A[i] nd A[j]

end while

swap A[s] nd A[j]

return j; // Return the index of pivot after the partition

Let A = {11, 7, 6, 48, 30, 12, 75}, and assume we call Quicksort(A, 0, 6). Show what happens during the first invocation of Partition. What is the value of p returned, and what are the two recursive calls made?

**Note: Credit will not be given only for answers - show all your work:**

**(5 points) steps you took to get your answer.**

11, 7, 6, 48, 30, 12, 75

J stops at 6, I stops at 48

(7, 6) 11 (48, 30, 12, 75)

**(5 points) your answer.**

P = 2

The two recursive calls made are to quicksort the left array (7, 6) and to quicksort the right array (48, 30, 12, 75)

1. Which is the running time of calling Mergesort on a linked list?

You must justify your answer with 1-2 lines of explanation.

**(2 points) your answer.**

O(nlogn)

**(3 points) justification for your choice.**

T(n) = O(n) + 2 \* T(n/2)

T(n/2) = O(n/2) + 2 \* T(n/4)

T(n) = O(n) + O(n) + 4 \* T(n/4)

This pattern happened O(logn) times.

Multiplied by n is O(nlogn)

**Section 2: Java Implementation**

1. Implement Problem 3 in Java.

Note:

Find a file called LinkedList.java in assignment 10 folder.

Complete the method of Listsort().

Test your method in the main method provided following the comments.

**Important: In all of the assignments of this course, when you are asked to implement an algorithm for a problem, your code will be evaluated based on:**

**5 points - Execution**

**Each file must run without error or warning on valid input described in the main method provided.**

**5 points - Within Code Documentation**

**Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?**

**20 points - Correctness**

**Is the algorithm implemented correctly? Does your method pass the test?**

**Note: Full credit (30 points) will be awarded for an algorithm that is O(nlog n). Algorithms that are O(n^2) slower will be scored out of 10 points.**

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt/.pdf), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName\_LastName\_assignment\_10'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**