CS3383, Winter 2019 Assignment # 3

Rasoul Shahsavarifar

Faculty of Computer Science, UNB

Due time: Friday, Feb/1/2019, 13:20 p.m

Student's full name: ..... Student ID:.....

Note:

- No submission after the due time will be accepted.
- The full credit will be given only for correct solutions that are described clearly.

Question 1 (5 marks) (Exercise 2.4 of DPU textbook): Suppose you are choosing between the following three algorithms:

- $Alg1$ solves problems by dividing them into five sub-problems of half the size, recursively solve each sub-problem, and then combining the solutions in linear time.
- $Alg2$ solves problems of size n by recursively solving two sub-problems of size $(n-1)$ and then combinig the solution in constant time.
- $Alg3$ solves problems of size n by dividing them into 7 sub-problems of size $n/3$, recursively solve each sub-problem, and then combining the solutions in $O(n^2)$.

By comparing the running time of these three algorithms (in big-O notation), which one you choose? Why?

Question 2 (7 marks) (modified exercise 2.14 of DPU textbook): You are given an array of n elements and you noticed that some of these elements are duplicate; that is they appear more than once in the array. Show that by doing some preprocessing work on the array, you can remove all duplicates from the array in $O(n \log n)$ time. Write your algorithm in pseudocode and analyze the running time of your algorithm. What is your expectation about the running time of an algorithm to remove the duplicates without doing the preprocessing work?

Question 3 (8 marks) (modified exercise 2.17 of DPU textbook):

a) Given a sorted array of distinct integers $A[1, ..., n]$, you want to find out whether there is an index i for which, $A[i] = i$. Give a divide and conqure algorithm in pseudocode that runs in $O(\log n)$.

b) Given a sorted array B, suppose that p and q are two elements of B. Design a divide and conqure algorithm to compute the number of elements between p and q (let $p \geq q$). what is the running time of your algorithm?

Bonus Question[1] (5 marks) You are given two sorted lists of size m and n.
Give an $(\log m + \log n)$ time algorithm for computing the k^{th} smallest element in the union of two lists.

[1] Solving the bonus questions provided in some assignments helps you to improve your overall marks on the current and previous assignments.