Home Work #7

DUE: 10am Monday Oct 12, 2020

Owing to the imminent mid-semester exam, no submission will be accepted after 10am Wednesday Oct 14, 2020

(upload portrait-mode PDF on Canvas)

☞ Handwritten assignments will not be accepted.

Start your assignment with the following text provided you can honestly agree with it.

- I certify that every answer in this assignment is the result of my own work; that I have neither copied off the Internet nor from any one else's work; and I have not shared my answers or attempts at answers with anyone else.

1. QuickSortLowest-k(A, p, r, k) is almost like QuickSort(A, p, r) (the algorithm we have seen in class). There are two differences. First, it has an extra input k, a positive integer. Second, it only needs to sort (in non-decreasing order) the lowest k elements, leaving the rest of the input in any order.

   Assume that the initial call is QuickSortLowest-k$(A, 1, n, k)$ and $1 \leq k \leq n$.

   While it is possible to ignore k and let the algorithm execute QuickSort, your algorithm should do better.

   Write pseudocode for QuickSortLowest-k(A, p, q, k) with comments that explain your strategy. (Use the notation we have used in class and is in the text. Make use of the Partition algorithm we have seen in class.)

   Explain why your algorithm is better.

2. What is the smallest change you would make to CountingSort (the algorithm we have seen in class) to reverse the sorting order (i.e., to make it sort in non-increasing order) without impacting either stability or execution time?

   (a) Explain why your changed algorithm would work.

   (b) Explain why your changed algorithm would preserve stability.

   (c) Explain why your changed algorithm would preserve execution time.