CSE 310 Assignment #4
(Max. Points: 30)

Due on: Friday, Oct. 9, 2020, 11:59pm Arizona time

General Instructions:

- This is an individual assignment, please do not collaborate. If there's programming part, make sure that you write every line of your own code. Using code written by someone else will be considered a violation of the academic integrity and will result in a report sent to the Dean's office.
- For all written exercises: **your answer should be clearly typed or written and must be saved in .pdf or .jpg format**. Note: **unreadable answer receives no credits!**
- All assignments must be submitted through the link posted on Blackboard, we do NOT accept any hand-in submissions or submissions sent through emails!
- Submission link will be closed automatically once the due date/time is past and **no late assignment will be accepted**. You will be allowed 3 times to submit the assignment before the due date/time, but we will only grade your last submission.

Objectives

- Heap & operations on a heap.
- Counting Sort
- Selection algorithms
- Algorithm analysis

Questions

1. [4 pts] Write pseudo codes for the function ***Change*(*A, heapsize, i, newValue*)** that accesses an element at the index *i* in the max-heap ***A*** (Note that the index of this heap starts from 1), and changes its value to the value of the parameter *newValue*, and also maintains its max-heap property. Note that *newValue* can be larger or smaller than the value of *A*[*i*]. You can use the heap functions/algorithms we discussed in class to implement this function. Also if the value of *i* is out of the range, then it should not change anything in the heap.

```
void Change(A, heapsize, i, newValue)
```

2. [4 pts] Suppose we use **RANDOMZIED-SELECT(*A, p, r, i*)** algorithm (pp. 215 Section 9.2) to select the minimum element of the array $A = \{3, 2, 9, 0, 7, 5, 4, 8, 6, 1\}$. Describe a sequence of partitions that results in a worst-case performance of RANDOMIZED-SELECT.

3. [4 pts] In the algorithm we learned in class **SELECT(*A, i*)**, the input elements are divided into groups of 5. Will the algorithm work in linear time if they are divided into groups of 7? Argue that SELECT does not run in linear time if groups of 3 are used.

4. [12 pts] Given an unsorted array of n unique integers, you need to return the k smallest of them in sorted order. You come up with three algorithms to solve this problem: They are:

- Algorithm **A**: Sort the array in increasing order, and list the first k integers.
- Algorithm **B**: Build a min-heap from these n integers, and then call Extract-Min k times.
- Algorithm **C**: Use the linear time selection algorithm to find the kth smallest integer, then partition the array about that number, and finally sort these k smallest numbers.

1) [3 pts] Let $T_A(n, k)$ denote the worst-case running time of Algorithm A. Analyze $T_A(n, k)$ using the big-O notation, in terms of n and k. Justify your answer.

2) [3 pts] Let $T_B(n, k)$ denote the worst-case running time of Algorithm B. Analyze $T_B(n, k)$ using the big-O notation, in terms of n and k. Justify your answer.

3) [3 pts] Let $T_C(n, k)$ denote the worst-case running time of Algorithm C. Analyze $T_C(n, k)$ using the big-O notation, in terms of n and k. Justify your answer.

4) [3 pts] Based on your analysis in the above, which algorithm would you choose to find the k smallest integers in sorted order, and why?

5. [6 pts] Use Figure 8.2 (your textbook pp.195) as a model, illustrate the operation of COUNTING-SORT on the array $A = \{6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2\}$. Show clearly array contents for B and C.