**Csci 41: Introduction to Data Structures**

**Lab Exercise 8**

Lab Instructor: Alex Liu

April 23, 2019

Deadline: Please submit your work at the end of this lab.

(**Turn in whatever you have at the end of lab. Continue to work on the lab exercise.   
Submit your final updated one by next lab**)

Objectives:

1. Priority Queue implemented by 3 different approaches
   1. Unordered array/vector
   2. Ordered array/vector
   3. Heap (Please continue to work on it at home. We will cover it either today or Thursday)
2. Dijkstra 3-way partition (from QuickSort lecture)

Directions:

1. [up to 5 mins] Read the descriptions of lab exercise.
2. [up to 25 mins] Discuss algorithms and coding of the lab exercise may be with teammates
3. [up to 1 hr] Code your lab exercise individually. You may ask for help from teammates or TA. But try to limit it unless you really get stuck.
4. [up to 30 mins] Review source code your teammates. Try to identify defects of your teammates. Return back the source code to teammates. Fix the defects that pointed out by your teammates.
5. [up to 1 mins] **You must** upload a single zip file (e.g., YourLastName-Lab8.zip) or a single CPP file to blackboard. The file should contain the following:
6. All \*.cpp and \*h files, if any
7. For each function or algorithm of your source code, you are requested to explain/comment the concepts/philosophy/theories of the function/algorithm. IF YOU DON’T HAVE DETAILED ENOUGH EXPLINATION FOR EACH FUNCTION/ALGORITHM YOU WILL GET 10 PTS OFF).
8. If you cannot finish this lab exercise, please continue to finish it before next lab.
9. Please DO NOT submit the entire Visual Studio projects or DevC++ file to me (**DO NOT SUBMIT EXE FILE TO ME. DO NOT SUBMIT PROJECT FILE TO ME**).

Exercise Summary:

Implement maximum priority queue (maxPQ) class with constructor, insert, delMax and isEmpty functions. There should be three maxPQ classes implemented using unordered array, ordered array, and heap, respectively. It is your decision to implement Template based maxPQ or not.



Figure 1. maxPQ and its functions.



Figure 2. Animation of maxPQ using unordered and ordered arrays.



Figure 3. maxPQ using heap – insert function



Figure 4. maxPQ using heap – delMax function