Lab:

Sorting algorithms

Prelab Task:

Please review the following sorting algorithms you have learned in COSC 120 & 220. You should review the design, implementation, and complexity of each algorithm.

- Bubble sort
- Selection sort
- Insertion sort
- Merge sort
- Quicksort
- Radix sort

Lab Task:

In this lab, we will implement a variation of selection sort algorithm. The double-ended selection sort starts from two elements and searches the entire list until it finds the minimum value and maximum value. The sort places the minimum value in the first place and maximum value in the last place, selects the second and second last element and searches for the second smallest and largest element. This process continues until the complete list is sorted.

- Declare and define the function deSelsort in a file named **deSelsort.h**
- Write your program in a file named **lab01.cpp** to sort an array, for example {13, 5, 2, 25, 47, 17, 8, 21}, so that your program can print the order of elements in the array after each pass of the double-ended selection sort and final sorting result.
- Run your program and print out the outputs
- Answer the questions: (1) When does your algorithm end (which means that the elements are all sorted in the array)? (2) What is the time complexity of the double-ended selection sort? Is it better than the selection sort?

What to Turn In

- Upload the following files to myClasses:
  - **deSelsort.h**
  - **lab01.cpp**
  - A **lab report** including your **answers** to the lab and at least two sets of **sample outputs** (a readable/clear screen shot)