Csci 41: Introduction to Data Structures

Lab Exercise 5

Lab Instructor: Alex Liu

April 2, 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. Elementary sorting algorithms

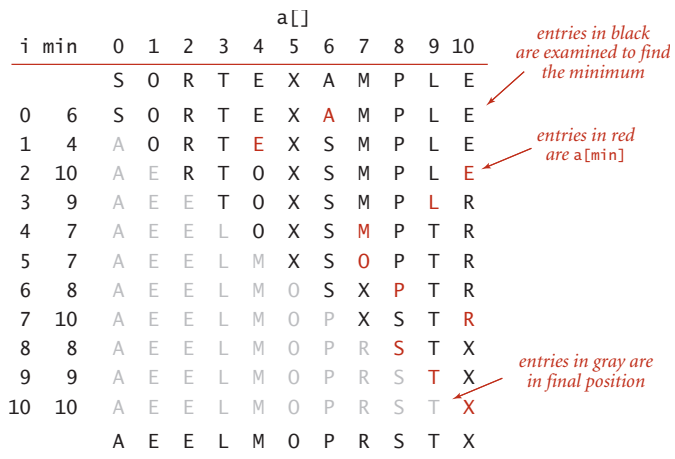
Directions:

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-Lab6.zip) or a single CPP file to blackboard. The file should contain the following:
   a. All *.cpp and *h files, if any
   b. 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).
6. If you cannot finish this lab exercise, please continue to finish it before next lab.
7. 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:

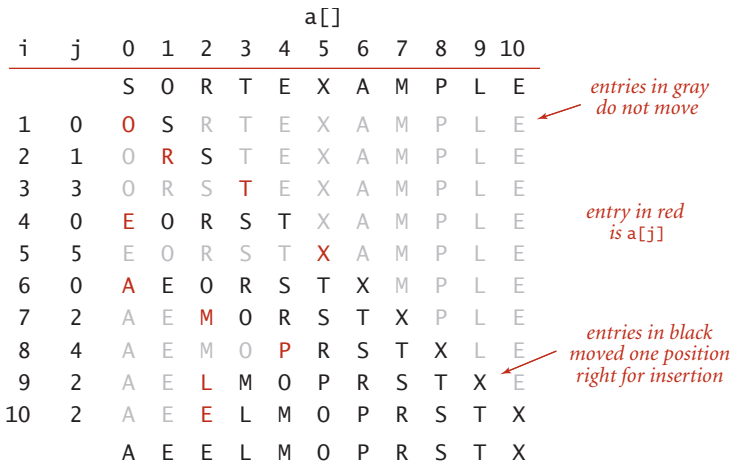
1. Read Section 1.4 page 12-17 about how to record execution time of an algorithms. Use the following diagrams to convert the algorithms into source code. (please note please try not to read the answer from lecture slides. This is an exercise to train you convert from animation to source code). Then record execution time of each algorithm.

1. Selection sort



Trace of selection sort (array contents just after each exchange)

2. Insertion sort



Trace of insertion sort (array contents just after each insertion)

3. Shell sort

input

S	O	R	T	E	X	A	M	P	L	E

7-sort

S	O	R	T	E	X	A	M	P	L	E
M	O	R	T	E	X	A	S	P	L	E
M	O	R	T	E	X	A	S	P	L	E
M	O	L	T	E	X	A	S	P	R	E
M	O	L	E	E	X	A	S	P	R	T

3-sort

M	O	L	E	E	X	A	S	P	R	T
E	O	L	M	E	X	A	S	P	R	T
E	E	L	M	O	X	A	S	P	R	T
E	E	L	M	O	X	A	S	P	R	T
A	E	L	E	O	X	M	S	P	R	T
A	E	L	E	O	X	M	S	P	R	T
A	E	L	E	O	P	M	S	X	R	T
A	E	L	E	O	P	M	S	X	R	T
A	E	L	E	O	P	M	S	X	R	T

1-sort

A	E	L	E	O	P	M	S	X	R	T
A	E	L	E	O	P	M	S	X	R	T
A	E	L	E	O	P	M	S	X	R	T
A	E	E	L	O	P	M	S	X	R	T
A	E	E	L	O	P	M	S	X	R	T
A	E	E	L	O	P	M	S	X	R	T
A	E	E	L	M	O	P	S	X	R	T
A	E	E	L	M	O	P	S	X	R	T
A	E	E	L	M	O	P	S	X	R	T
A	E	E	L	M	O	P	R	S	X	T
A	E	E	L	M	O	P	R	S	T	X

result

A	E	E	L	M	O	P	R	S	T	X