CS 220 – Data Structures and System Programming – Spring 2020

PEX 2 –Sort Analysis – 75 Points

Electronic Submission Due @ 2300 on Lesson 11

M-Day: Wed, 5 Feb2020
T-Day: Thurs, 6 Feb 2020

Help Policy:

AUTHORIZED RESOURCES: Any, except another cadet's work.
NOTE:

- Never copy another person's work and submit it as your own.
- Do not jointly create a program unless explicitly allowed.
- You must document all help received from sources other than your instructor or instructor-provided course materials (including your textbook).

Documentation Policy:

- You must document all help received from any source other than your instructor or instructor-provided materials, including your textbook (unless directly quoting or paraphrasing).
- The documentation statement must explicitly describe WHAT assistance was provided, WHERE on the assignment the assistance was provided, and WHO provided the assistance, and HOW it was used in completing the assignment.
- If no help was received on this assignment, the documentation statement must state "None."
- If you checked answers with anyone, you must document with whom on which problems. You must document whether or not you made any changes, and if you did make changes you must document the problems you changed and the reasons why.
- **Vague documentation statements must be corrected before the assignment will be graded and will result in a 5% deduction on the assignment.**

Turn-in Policies:

- On-time turn-in is at the specific day and time listed above.
- Late penalty is a 50% cap in the points you can earn on the PEX. For example the most you could earn on a 100 point PEX is 50 points. Late PEXs can be turned in up to 1 week late.

OBJECTIVES

Upon completion of this programming exercise, students will be able to:

- Perform an empirical analysis of an algorithm in order to determine the physical time and number of operations it takes to execute on varied data.
- Infer Big-O complexity of an algorithm based on analysis.
- Identify various sorting algorithms based on run-time analysis

1. DESCRIPTION

For this exercise you will analyze the behavior of sorting algorithms.

To perform an empirical analysis of an algorithm, you do the following:

- Time how long it takes for an algorithm to execute on data structures of various sizes and types (random, sorted ascending, and sorted descending).
- Examine the run-time behavior of different sorting algorithms
- Use Excel to plot the size-time pairs in a scatter plot. The x-axis is the data size. The y-axis is the associated run-time. The resulting plot describes how the run-time behavior of an algorithm changes as the size of the data structure grows large.

2. BACKGROUND

While working on a recent program I accidently lost track of some of the implemented algorithms and need your help to figure out which algorithm was implemented. The algorithms are listed as Lost_Sort_A through Lost_Sort_F in the PEX2 executable: AlgorithmVisualization_Student.exe. You will need to run through the various analysis routines to identify the Big-O time complexity for each of the lost sorts for different data sizes and initial data orders (Ascending, Descending, and Random).

3. REQUIREMENTS

3.1 LOST ALGORITHMS

The lost algorithms are some ordering of the Selection Sort, Insertion Sort, Bubble Sort, Shell Sort, Merge Sort, and Quick Sort. You will need to identify the algorithm that corresponds to each of the Lost_Sort implementations.

3.2 TESTING AND DATA GATHERING

There are two parts to the AlgorithmVisualization_Student tool that should help you identify the algorithms of each of the Lost_Sort implementations. The first part is the live algorithm graphing chart on the main window. You should use this display to help compare the sorting actions with what you would expect based on the actual sorting algorithms. Unfortunately this display will not be sufficient for you to justify a claim of an algorithm.

To fully identify each algorithm you will need to do some analysis on the runtime performance. To do so you will use the Analysis window (reached by clicking the button “Analyze” from the main page. This window contains the following options:

- Maximum Data - the size of the largest data set to be generated
- Number of Runs - how many different data sets will be generated and sorted up to and including the Maximum Data amount of elements
- Initialized Order -whether the initial data is generated in Ascending, Descending, or Random order

Clicking “Run” prompts the tool to generate the specified data elements, run the selected sorting algorithm, and return the execution time and number of operations required to sort the data elements. The tool provides a table displaying the execution time (in milliseconds), and the number of operations performed for each of the dataset sizes generated.

NOTE: As some of these algorithms are faster than others, you should start with a lower Maximum Data value and then increase the value until you find a Maximum Data value that allows you to accurately assess the algorithm’s execution.

3.3 Analysis

The first step of this effort will be to analyze each of the Lost_Sorts to find the Big-O and thereby help determine which type of sort is being implemented. You must answer the below questions:

- What is the Big-O for the algorithm when the data is initialized in ascending, descending, and random order?
- Which algorithm do you believe the Lost_Sort is executing?
- Explain why you believe this algorithm is being executed?

For this part of the PEX, record your answers to the questions, then enter them on the CS220 website in Canvas for submission credit. Your answers should be well-written to include proper punctuation and grammar, and they should be of sufficient length to properly answer the question.

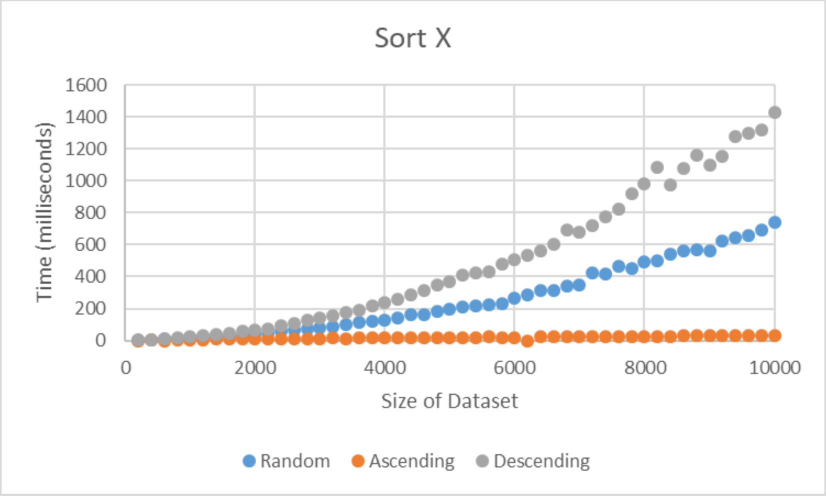
NOTE: Canvas will only allow you to submit once though it will allow you to exit and continue while autosaving your answers. Once you click "Submit" the questions will be locked and not retrievable.

3.4 Charts Analyzing Each Sort

In addition to your analysis for each of the Lost_Sorts on Canvas you must also submit a word document uploaded to the CS220 website on Canvas. This document must include charts for each of the Lost_Sorts supporting your Big-O analysis. The charts must each be labelled with a title referencing which Lost_Sort you are charting, a legend to indicate which chart items are displaying your Ascending, Descending, and Random data, and axis labels. You are encouraged to reference these charts in your analysis statements via Canvas.

4. Helpful Hints

- Read and re-read the write-up as required.
- Google "scatter plots with trend lines in ms excel" to uncover how to plot your data.
- The timing for the algorithms is based on the runtime speed of the execution on your computer. If you are running various other processes while conducting the test then you will see variations in your charts that will hamper your ability to accurately analyze the algorithms.
- Below is a sample chart done in Excel for one of the sorts using random, ascending sorted data, and descending sorted data.



5. Submission Requirements

- There are two deliverables you will need to submit for this assignment:
    - The questions linked to the PEX2 assignment on the course Canvas site
    - A word document with the charts you generated for each of the sorting algorithms and a written discussion comparing the sorting algorithms.
        - You must clearly label which sorting algorithm is illustrated.
        - Please note, there will be points for the quality of content and format.
- Documentation statements must be posted with the submission and included on your word document.

6. Questions from Canvas Quiz

Below are the questions that are posted on the Canvas Quiz. Please note, you will only be able to submit once to the Canvas quiz. You should prepare your answers prior to starting the quiz.

6.1 Questions for each of the Lost_Sorts (repeated for Lost_Sort_A through Lost_Sort_F):

- (2 pt) What is the Big-O for the Lost_Sort when the data is initially in Ascending Order? (free form text)
- (2 pt) What is the Big-O for the Lost_Sort when the data is initially in Descending Order? (free form text)
- (2 pt) What is the Big-O for the Lost_Sort when the data is initially in Random Order? (free form text)
- (1.5 pt) What Sorting Algorithm do you think this Lost_Sort is executing? (Insertion, Selection, Bubble, Merge, Shell, or Quick)
- (4 pts) Justify why the Lost_Sort is your hypothesized sorting algorithm. In your discussion address any outliers from your charts. Your discussion must address the Big-O calculated for the different data set initializations.

6.2 Questions for the overall analysis

- (1 pts) You likely saw that the measurement of the algorithms' performance was not precise. Discuss why the measured values used in this exercise are approximations for Big-O and not a direct measurement.
- (5 pts) Based on your analysis, compare and contrast the different sorting algorithms and provide a justification for when you may/would use different algorithms for solving particular sorting requirements.

PEX 2 Grade Sheet

Name:________________________________

	Points	
Criteria	Avail	Earned
Analysis including the Canvas questions and completed charts for each of the unknown sorts.		
Lost_Sort_A	**11.5**	
Lost_Sort_B	**11.5**	
Lost_Sort_C	**11.5**	
Lost_Sort_D	**11.5**	
Lost_Sort_E	**11.5**	
Lost_Sort_F	**11.5**	
Comparison of Sorts	**10**	
Subtotal	**75**	
Submission requirements not followed (-5%)	**-4**	
Documentation statement missing (-5%)	**-4**	
Late penalties – 50% Cap (may turn in 1 week later than due date / time)	**max score= 37.5/75**	
Total	**75**	

Comments: