CSI 403 Data Structures and Algorithms Spring 2020
Programming Assignment I

Date given: January. 27, 2020 Due date: February 12, 2020

The deadline for this assignment is 11:59 PM, February 12, 2020. The assignment will not be accepted after the deadline.

Student learning outcomes:

SLO1. Analyze the complexity of simple algorithms and discuss advantages of alternative algorithms for the same problem.

SLO2. Recognize design patterns such as greedy, divide-and-conquer and others.

Very important note: You MUST use 3 java Files and solve the assignment using the Driver and Point classes. Submissions using other data structures and/or additional classes or libraries will be scored as 0 points.

You must turn in electronically:

1) A screencast URL using Jing that contains the following:
    a. Scroll through all of your source code slow enough to read.
    b. The source code compiling
    c. The source code running
2) The three java files with implementations
3) A PDF or MS WORD file that shows your analysis in part (b).

(It is not a requirement that you talk on your screencast)

Java Files to be used in the assignment:

1) Point.java: defines the Point object and some comparators to support sorting. Do not modify this file. THIS FILE IS GIVEN
2) ClosestPair.java: Create method stubs for the two algorithms to be implemented. THIS FILE IS GIVEN BUT NEEDS MODIFICATION. (see comments in file)
3) ClosestPairDriver.java: Contains tests to verify the correctness of your algorithms as well as a main() method. You need to implement the timing experiments required for part (b) of the assignment (see below) THIS FILE IS GIVEN BUT NEEDS MODIFICATION. (see comments in file)

The objective of this programming assignment is to implement two different algorithms for the closest pair problem and experimentally compare their computation times.

In the closest pair problem, we are given a set of n points in the plane. Each point is specified by its x and y coordinates, which are real numbers. Recall that given two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$, the (Euclidean) distance $d(p, q)$ between them is given by the formula

$$d(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$$

The goal of the closest pair problem is to find the shortest distance between any pair of points in the set.

CSI 403 Data Structures and Algorithms Spring 2020
Programming Assignment I

A simple (brute-force) method of computing the shortest distance is to compute the $n\ (n - 1)/2$ distances between all pairs of points and take the smallest of these distance values. Clearly, this algorithm runs in $\Theta(n^2)$ time. For this problem, there is a divide-and-conquer algorithm that runs in $O(n \log n)$ time. A description of this algorithm was presented in class (see lecture slides, also Chapter 33.4 of the book). You are required to implement both the brute-force algorithm and the divide-and-conquer algorithm and determine the actual times used by these two algorithms on test inputs. There are two parts to this assignment.

Part (a): Implement the *Brute Force* and *Divide-And-Conquer* algorithms. Use the ClosetPair.java file to hold your methods. Consult the lecture slides for the pseudocode and analysis justifying the $\Theta(n^2)$ and $O(n \log n)$ growth rates respectively. Beyond compilable, all tests should be defined in the driver class should pass with success. Please document your justification of the test you used in comments.

Part (b): In this part, you are required to experiment with inputs of sizes 10,100,1000 and 10000 by employing the random point generator that you should implement in ClosestPairDriver.java. To do this analysis, you should implement a runnningTimeComparison() method in the ClosestPairDriver.java. Report the computation times of the two algorithms as discussed below.

Using the outputs produced by your algorithms, you should prepare a table with four rows (one corresponding to each size) and three columns labeled ***No. of points***, ***Time used by Brute Force*** and ***Time used by Divide-and-Conquer***. Plot this table (using MS Excel or any other software for plotting) in a line chart with x axis the number of points and y axis: time taken by an algorithm. You will have two curves: one for each of the algorithm. Plot also in the same figure the $f(n) = c_1n^2$ and $g(n) = c_2n \log n$ functions for the same range of number of points n = (10, 100, 1000, 10000), by choosing appropriate constants c_1 and c_2 to fit visually your observed running times. Discuss what you observe about those constants. Do they matter for small *n*? How about, large *n*?

The table, your explanation and plots and plots should be submitted as a separate PDF or MS word file (Do not use other formats. If you have to export to PDF before submitting).

What you will turn in to Blackboard no later than the deadline:

1) All source files in a zip file
2) A screencast URL or MP4 file that shows
    a. your code
    b. the compiling of the code
    c. the run time analysis of the code.
3) A 1-2 page document with the write up of the analysis

CSI 403 Data Structures and Algorithms Spring 2020
Programming Assignment I

Rubric

Item	UNSATISFACTORY	Developing	Satisfactory	EXEMPLARY
	0 points	2 points	3 points	4 points
Video Submitted	No video exists	Another method was used to submit the assignment	video exist but does not have all the elements	video exists and works properly
Performance Indicator #1: Design and implementation of the solution of the closest pair problem using two different algorithmic paradigms.	The program does not compile or program fails to read the provided test and produce an output.	Correct implementations of one of the two algorithms that achieve the correct result on the provided test cases.	Correct implementations of the two algorithms that achieve the correct result on the provided test cases.	Correct implementations of the two algorithms that achieve the correct result on all test cases.
Performance Indicator #2: The complexity analysis of both algorithms.	Incorrect analysis or no analysis.	Correct analysis one of the two algorithms.	Correct analysis of both the algorithms.	Correct analysis of both the algorithms with a comparison with appropriate justification.
Performance Indicator #3: The execution time analysis (in ms) of both algorithms.	Incorrect analysis or no analysis.	Correct analysis one of the two algorithms.	Correct analysis of both the algorithms.	Correct analysis of both the algorithms with a comparison with appropriate justification.
Part B: 4 test cases exist	No test cases were shown	One test case was shown	Two or three test cases were shown	All four test cases were shown
Proper C1 and C2 were chosen	No c1 and c2 were chosen	One or the other C value was chosen	Two C values were chosen but they are incorrect	Proper C values were chosen
Overall look and feel of the assignment	Submission is not formatted and does not show understanding of the topic	Understanding of the topic is evident but the solution is not well organized	Understanding of the topic exists and there are minor issues with the presentation of the findings	Solution is well formatted and make proper sense for the assignment