CS 3305 Data Structures

Assignment 05 – Big O Check D2L for Due Date (100 points total)

Note: Never hard-code test data in the test program, unless explicitly stated otherwise in the assignment. Always allow the user to enter the test data using a menu option.

GENERAL SUBMISSION REQUIREMENTS

Upload all files individually as specified, not as zip files, to Assignments in D2L. Do not email files.

Make sure your program compiles, runs and produces the correct output.

Ensure you have the correct file name(s), and author header, as specified in the Assignment.

Always use meaningful labels for prompts, inputs, and outputs.

Always use comments, indentation and whitespace as shown in examples.

Objectives The purpose of this lab is to reinforce concepts of runtime analysis (Big O) and sorting.

Assignment 05 PART 1 Runtime Analysis (25 points) - Note Part 1 is a separate deliverable:

1. Use Big O notation to estimate the time complexity of each of the following code fragments. Clearly show all steps of your work. Review examples in textbook pp. 822-827

    1.1 sum = 0;

```
for (int i = 0; i < n; i++){
sum++;
}
```

    1.2 sum = 0;

```
for (int i = 0; i < n; i++)
{
for (int j = 0; j < n; j++){
sum++;
}
}
```

1.3 sum = 0;

```
for (int i = 0; i < n; i++){
for (int j = 0; j < n * n; j++) {
sum++;
}
}
```

1.4 sum = 0;

```
for (int i = 0; i < n; i++) {
for (int j = 0; j < i; j++) {
sum++;
}
}
```

1.5 sum 0;

```
for (int i = 0; i < n; i++) {
for (int j = 0; j < i * i; j++) {
for (int k = 0; k < j; k++) {
sum++;
}
}
}
```

DELIVERABLE INSTRUCTIONS – PART 1

You do not need screenshots for A5-Part 1 and you do not need to copy and paste any code. If you work your problems out on paper, you may scan the work into a word doc or image. Phones have scan apps that do a good job. Name your file LastName-A5-Part-1-Runtime.docx or .pdf. Last step is to upload word/pdf file to D2L. To clarify, upload the following for A5-Part-1.

1. word/pdf doc that contains your work for problems 1.1 thru 1.5; and
2. you DO NOT NEED to upload a .java file for this part of the assignment.

Only your latest submission is kept so if you update a file and want to resubmit you must upload all files. Submit everything to the assignment submission folder in D2L by the due date posted in D2L.

No zip file or email submissions are accepted.

Late penalties of 10 % per day are in effect for this assignment.

Assignment 5 – PART 2 Program Analysis (25 points) :

Write a java program that will convert distance measurements from miles to kilometers for input of n numbers, and determine the worst case efficiency.

Specifically, the program needs to convert 0, 10, 20, 30, 40 , 50, 60, 70, 80, and 90 miles to

kilometers displaying both the number of miles and the number of kilometers on the same line.

Tip: 1 mile = 1.6 x km

```
// Name: <your name>
// Class: CS 3305/ put your section number after the /
// Term: Fall YYYY
// Instructor: Sharon Perry
// Assignment: 5-Part-2-Miles
```

DELIVERABLE INSTRUCTIONS – Part 2

Capture a **READABLE** screenshot(s) of your program output and paste into a word/pdf document. Readable means readable! Screenshots ***should not be an entire desktop*** – use some type of snipping tool. After your output screenshots, copy and paste the source code for your program into the word/pdf doc. Save doc as a file named LastName-A5-Part-2-Miles.docx or .pdf. Last step is to upload everything to D2L, to clarify, upload

3. word/pdf doc that contains output screenshot and copy/pasted source code; AND
4. your .java file that contains your source code.

Only your latest submission is kept so if you update a file and want to resubmit you must upload all files. Submit everything to the assignment submission folder in D2L by the due date posted in D2L.

No zip file or email submissions are accepted.

MAKE SURE YOUR CODE HAS COMMENTS ! We are getting submissions without comments in the code. No comments in source code = (-30) points *per each Part of the assignment.* This penalty grows as time goes on !!

Late penalties of 10 % per day are in effect for this assignment.

Assignment 5 PART 3 Sorting (50 points) - Note Part 3 is a separate deliverable:

You must write your own Mergesort code using Linked Lists, not ArrayLists, and you may use Linked List class available in the Java Library. You may not use any predefined sorting methods in the Java Library.

One of the advantages of Mergesort algorithm is that it can easily be adapted to sort a linked list of values. This is because the algorithm retrieves the values from the two lists being merged in the order that they occur in the lists. If the lists are linked lists, then that algorithm can simply move down the list node after node.

Write a program that sorts a linked list of integers using the Mergesort algorithm. The program will read the integers into a linked list, and then sort the linked list using Mergesort. This will require additional linked lists, but you should use linked lists, not arrays, for all your list storage.

Use the data provided in the text file attached to the assignment, called mergetext.txt. You may read the data from the mergetest.txt file that we have provided - or you may hard code that data, into one linked list. Once you have the linked list, then you begin the Mergesort algorithm, and it takes the values and splits them and merges them in pairs, then merges the pairs, etc until the correct order has been achieved..

Do not forget to include author header in each submitted file as shown, no header, no points!

```
// Name: <your name>
// Class: CS 3305/ put your section number after the /
// Term: Fall 2022
// Instructor: Sharon Perry
// Assignment: 5-Part-3-Sorting
```

DELIVERABLE INSTRUCTIONS – Part 3

Capture a **READABLE** screenshot(s) of your program output and paste into a word/pdf document. Readable means readable! Screenshots ***should not be an entire desktop*** – use some type of snipping tool. After your output screenshots, copy and paste the source code for your program into the word/pdf doc. Save doc as a file named LastName-A5-Part-3-Sorting.docx or .pdf. Last step is to upload everything to D2L, to clarify upload

1. word/pdf doc that contains output screenshot and copy/pasted source code; AND
2. your .java file that contains your source code.

Only your latest submission is kept so if you update a file and want to resubmit you must upload all files. Submit everything to the assignment submission folder in D2L by the due date posted in D2L.

No zip file or email submissions are accepted.

MAKE SURE YOUR CODE HAS COMMENTS ! We are getting submissions without comments in the code. No comments in source code = (-30) points *per each Part of the assignment*. This penalty grows as time goes on !!

Late penalties of 10 % per day are in effect for this assignment.