

Algorithms and Advanced Programming Continuous Assessment

Module Name: Algorithms and Advanced Programming (AAP)

Academic Year: 2021/22, Semester III

Programs: HDSDEV_JAN, HDSDEV_INT, HDCSDEV_INTJAN22

Total Marks: 100 marks with 50% contribution to the total module result.

Submission Deadline: Week 11 (02-08-2022 @23:59)

Project Background:

A hypothetical Irish climate change advocacy group wants to analyse the carbon footprint (CFP) of higher education students in Ireland. The group collected data about all higher education students concerning their transport usage to travel to/from their respective universities. So far, the group has managed to collect 10,000 CFP records containing 8-digit StudentCode (integer), studentName (string), ModeOfTransport (Bus, Train, Car, Motor Bike, Bike, E-scooter, Walk), distance in KM (integer) and Emission in KG (float). The data is already sorted by studentCode in ascending order.

Assuming that you are part of development team, your responsibility includes analysis of the data and writing program fragments that will manage information related to the CFP. Now you are assigned to implement and generate report for the following sprints before 02-08-2022 @23:59. A supplementary Java code that will read and store the dataset in an array of CFP, and the dataset will be provided for you on the Moodle page.

Part 1: Sorting and Searching: Algorithm Analysis

1. Write a Java method(s) that sort the CFP information using a Merge-sort algorithm. The CFP should be sorted by its studentName in ascending order. **(30 Marks)**

Input: 10,000 unsorted CFP records,

Output: 10,000 CFP records sorted in ascending order by studentName.

2. Write a program that tracks the time of your sorting algorithms in Q1 by running your program and tracking the average elapsed time. Your experiment should take an input of 10, 100, 1000, 10000 records for the sorting algorithm. Use the following template to track the time and generate a graph of the growth of the algorithm. **(20 Marks)**

Input size	10	100	1000	10000
Merge Sort				

3. Write an implementation of a binary search using Java method(s) that search a record by the StudentCode field. **(20 Marks)**

Input: StudentCode

Output: the CFP record/s corresponding to the StudentCode.

Part 2: Defensive Programming and Exception Handling (New: Implement this section only for a data entry for new CFP details)

1. Transportation mode should always be one of the values (Bus, Train, Car, Motor Bike, Bike, Walk), Write a Java setter method (setModeOfTransport) that checks the value of the ModeOfTransport and throws a custom exception with a message 'Unsupported Mode of Transport. Please correct the transportation mode by choosing from (Bus, Train, Car, Motor Bike, Bike, E-scooter, Walk)'. **(15 Marks)**
2. Write a Java method that checks the correct values of the distance field. The minimum distance in KM is 1 and the maximum is 500. **(15 Marks)**

Example Invalid input: distance = 0.25 (distance below trackable limit)

Example Invalid input: distance = 12570 (Invalid distance limit for Ireland)

Available Resources:

1. A csv file containing 10,000 records is available for download from module's Moodle page.
2. A Java snippet to read the csv file will be made available. You may use the code to load your data, or you may build your own way of reading the records from the csv file.
3. You may use a suitable data structure to represent the CFP information.

4. You may use any Java IDE to complete your assignment.

Submission Checklist:

1. A pdf/word document containing the answers to each of the above questions. Include any textual description, justification and assumptions along with your answers.
2. A Java source file containing the implementation for the questions. Write your methods to the respective classes and provide the question number as a comment on top of your methods. Java compiled class will not be accepted.
3. Make sure all your code compiles correctly and all required packages are imported correctly.

Submission Guideline:

The submission link will have two parts.

Part 1. Document upload (upload your pdf/word file here)

Part 2. Java code upload (upload a single zip file containing your classes and methods)

Naming Convention:

[YourID]_[YourProgramCode]_[YourLastName]_[YourFirstName]_Description.doc/pdf

[YourID]_[YourProgramCode]_[YourLastName]_[YourFirstName]_AAP.Zip

Example:

- X123456_HDSDEV_SEP1_Abgaz_Yalemisew_AAp_Description.doc
- X123456_HDSDEV_SEP1_Abgaz_Yalemisew_AAP_Code.zip

Submission Deadline: 02-08-2022 @23:59

Marking Rubrics: Marking rubrics will be uploaded on your Moodle page.