**Algorithms and Advanced Programming Continuous Assessment**

**Module Name:** Algorithms and Advanced Programming (AAP)

**Academic Year:** 2024/25, Semester II

**Programmes:** HDSDEV\_SEP, HDCSDEV\_INT, HDCSDEV\_J22\_Y1\_O

50% contribution to the final result.

**Submission Deadline:** March 26th 23:55

**Learning Outcomes:**

|  |  |
| --- | --- |
| LO1 | Use iterative and recursive techniques to design and implement sorting and searching algorithms. |
| LO2 | Conduct in depth algorithm analysis in terms of time complexity and present the results of analysis. |
| LO3 | Evaluate algorithms, identify an algorithm from a range of possible options, and implement the algorithm to solve computational problems in particular contexts. |
| LO4 | Identify and apply best practices including exception handling and design patterns in the implementation of software solutions to solve real-world problems. |

**Project Background (50%):**

Students may choose between completing part 1 of this assignment and a short quiz (10 questions), or to not submit part 1 and complete a long quiz instead (20 questions). Should students choose to complete part 1 of this assignment, they will receive half the marks for part 1 and half the marks for part 2 (the quiz). Otherwise, the full CA grade will be based on the long quiz only.

**Part 1 dataset description:**

Each student is expected to complete the following tasks for the following sets of data. Each dataset contains 6 columns and 10,000 rows. Your assigned dataset is dependent on your student number. Match each part of your student number to the relevant descriptions of what dataset you should use and how you should sort, transform, and retrieve your data.

**Note:** the development of the software should be your own. It is okay to use samples from the lectures and labs, but you should highlight the parts of development that are your own. You should not use code developed by other students also completing the assessment. I will not accept algorithms that are imported from some existing library in java (i.e., all sorting should be done using the basic C-style array, not an array list, and the sorting algorithms should be implemented by the student themselves, **not called as a method from an imported library**). Please do not use approaches not covered in class, such as the application of lambda functions.

**Choosing your dataset**

Your dataset will be chosen based on the final digit of your student number. The table below highlights what number is associated with what dataset. It also highlights the columns in each dataset.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| last digit | Dataset file name | Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 |
| 0  3  6  **9** | Staff.csv | emp\_no (integer) | first\_name (String) | second\_name (String) | department (String) | wage (float) | project\_completion\_rate (int) |
| 1  4  7 | Film.csv | film\_id (integer) | genre (String) | director\_name (String) | title (String) | length (float) | rating (float) |
| 2  5  8 | Stock.csv | stock\_no (integer) | product\_size (float) | profit (float) | product\_type (one of 3: food/furniture/ clothing) | weight (float) | product\_name (String) |

**Part 1: Sorting and Searching: Algorithm Analysis**

1. Write a sorting algorithm that sorts the data using a ***column from your selected dataset based on your student number (as described above)***. For the column selection, you can use the table given below. You may choose any efficient sorting algorithm we covered in this module. If two items have the same value for your selected columns, then sort them based on Column 1, which has unique values.

You have been provided with additional resource files to assist you in reading CSV data. These files are ReadFilmData.java, ReadStaffData.java, and ReadStockData.java. Based on your selection of datasets, you can use any of these files to read the records from the CSV files.

You will receive better marks for optimal (low run-time) solutions. Highlight in the submission the reason why you chose your sorting algorithm **with reference to the run-time complexity**. The sorting algorithm must be your own implementation. You will receive 0 marks for using an imported library to complete this task.

**(10 Marks)**

***Input:*** 10,000 unsorted records,

***Output:*** 10,000 records sorted in ascending order by column X

|  |  |
| --- | --- |
| Second last digit of student number | Column to sort on |
| 4  5 | 2 |
| 3  9 | 3 |
| 1  7 | 4 |
| 2  8 | 5 |
| 0  6 | 6 |

1. Experimentally analyse the time complexity of your sorting algorithm you write for question 1 above. Show your results by taking the average elapsed time for 10, 100, 1000, 5000 and 10000 records. **(15 marks)**
2. Write a searching algorithm that accepts the sorted column type and searches the data record from the with your dataset. You should select data based on the description given in question 1. Note there could be multiple matches for a query. If no X is found, display “Not an existing X!” (Where X is the title of the sorted column like a film title, customer name, item name etc) message. If multiple items have that value return all matches. **(10 Marks)**

***Input:*** 10,000 sorted records,

***Output:*** list of record details with the same associated column label as entered by the user

1. Analyse and determine the time complexity of the searching algorithm you write in q3. **(15 Marks)**

**Part 2:**

The remaining 50 marks will be assessed using an in-class open-book quiz. The quiz will be composed of an array of short and multiple-choice questions. This quiz will take place during your usual lab time on the week of March 27th. During this quiz you may freely use the internet (excluding social media or sites/apps hosting AI chat services) and notes. You may not ask another person for the answer or query a large language model (AI/chat GPT). Those found to be in breach of this will receive an automatic zero on this quiz. Should you choose not to complete part 1 of this assignment your grade will be based on the long format version of the quiz (including 20 questions). On the quiz day you should be careful to choose the “short quiz” if you completed part 1 of this assignment, or the “long quiz” if you have not completed this section.

**Available Resources:**

1. A csv file containing 10,000 records is available for download from your 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 employee 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. It should contain any input-process-output descriptive diagrams to communicate some functions in the programs.
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]\_AAP\_Description.doc/pdf

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

**Example:**

* + X123456\_ HDSDEV\_INT \_Abgaz\_Yalemisew\_AAP\_Description.doc
  + X123456\_ HDSDEV\_INT \_Abgaz\_Yalemisew\_AAP\_Code.zip