Ton Duc Thang University
Faculty of Information Technology

FINAL EXAMINATION (ONLINE CLASSES)

COURSE: DATA STRUCTURES AND ALGORITHMS

SEMESTER I ACADEMIC YEAR 2021 - 2022

I. Examination regulations

- Students submit source code and a report that is written in Word.
- Students create a folder named **HoTen_MSSV**, contains the report in **pdf** format, and for each question there will be a corresponding folder that contains the **.java** code files. Compress the folder with the **.zip** format and submit the compressed file follow the instruction of your lecturer.
- Students do the report with the faculty report format.
- Submission date: **follow your lecturer notification**.
- **STUDENTS MUST NOT COPY OTHER STUDENTS IMPLEMENTATION.** If there are cheating found, all of the students involved will get 0 point and get disciplined by the disciplinary council.

II. Examination guidelines

- For questions that have the Theory part, students write the theory onto the report.
  STUDENTS MUST NOT COPY FROM THE INTERNET.
- For questions that have the Practical part, students write code according to those questions description. After finishing the implementation, students copy their implementation to the report.
- Note that when copying the source code to the report, students should format it for better readability (have indentation for block of code).
- For the **title** of the report, students write “Final examination report of Data Structures and Algorithms course”.
- **For all the students that copy content from the Internet if detected will get 0 point for this entire report.**
- **The schedule for the oral exam will be notify later on Google Classroom after students complete their submission.**

III. Oral exam points

For the points that are included in the questions below, half of these points are reserved for the oral exam. Students only get the full points of the question if their implementation are correct and they are able to answer the related questions of the lecturer.

IV. Examination questions

Question 1: Recursion (1 point)

All of the answers in Question 1 must be implemented by using recursion. For those that used loop, point will not be counted.

a. **(0.5 points – Theory)** Choose from each group one problem and describe its recursiveness (describe, draw the recursion tree with an example of at least 5 times of recursion):

i. Group 1:

• Calculate ��$^{��}$

• Find the sum of all of the digits of a positive integer.

ii. Group 2:

• Find the �� th fibonacci.

• Solve the combination problem (�� choose ��).

• Find the largest number in an array.

b. **(0.5 points – Practical)** Create a class to implement the code for 2 problems that you have chosen and create a main method to check your implementation.

Question 2: Sorting (1 point)

Students provide themselves with an array of 10 random integers (the position of the elements of the array must be random, must not fall in these cases: ordered array, one part of the array is ordered)

a. **(0.5 points – Theory)** Give an example describe each step of sorting the array using 02 in these 03 algorithms: Bubble Sort, Insertion Sort, Selection Sort. (Students must describe the changes of the elements in the array after each step).

b. **(0.5 points – Theory)** Give an example describe each step of sorting the array using 01 in these 02 algorithms: Merge Sort or Quick Sort. (Students must describe the changes of the elements in the array after each step).

Question 3: Stack (2 points)

a) **(1 point – Theory)** Describe the way using Stack to create a Reverse Polish Notation (RPN). This algorithm will transform an expression from infix notation to postfix notation. Give an example of an infix expression that has at least 6 mathematical operations (has all of the addition (+), minus (-), multiply (*) and division (/)) and at least 2 pairs of brackets, apply the algorithm to transform the expression to postfix notation.

b) **(1 point – Theory)** Describe the way using Stack to calculate the result of the postfix notation. Apply the algorithm to the postfix notation in part a).

Question 4: Data structures (3 points)

Theory (1.5 points):

Analyze the problem of managing students using Array, Linked List, or an AVL tree to store the data. Show the strength, weakness and compare the efficiency of each data structure on these operations:

a) (0.25 points) Add a new student

b) (0.25 points) Remove a student

c) (0.25 points) Search a student

Practical (1.5 points):

Implement the code to solve the problem of managing students using Linked List (students are not allowed to use Linked List API) and using AVL tree with the data given by students, at least 15 objects. For each Student object, it will have these attributes: student ID (8 characters, the first character can be a letter), student full name, cumulative GPA, and 2 additional attributes that students will give. Implement the methods for these operations on the two data structures. (For the AVL tree, use student ID as the key to comparing)

a) (0.25 points) Add a new student.

b) (0.25 points) Remove a student.

c) (0.25 points) Search a student.

Question 5: Graph (3 points)

Build a graph represent the users of an social network. Each User will have a user ID, username, age, gender and job.

Theory (1.5 points):

a) (0.25 points) Draw a graph with each vertex being the user ID, two users that are friend with each other will have an edge between their two vertices. The graph must have at least 30 vertices and 50 edges. The graph can contain some isolated vertices.

b) (0.5 points) Show the result of traversing the graph using BFS and DFS starting from a vertex which students choose.

Practical (1.5 points):

a) (0.25 points) Students create a text file, read the file and build the graph which students gave in Theory part (Students can use adjacency matrix, adjacency list, edge list to represent the graph).

b) (0.25 points) Implement a method return the friend list of a user passed as a parameter.

c) (0.25 points) Implement a method that count the mutual friends between two users passed as parameters.

---THE END---

If students have any questions, please contact email: tranthanhphuoc@tdtu.edu.vn and dungcamquang@tdtu.edu.vn