UNIVERSITY OF VAN LANG

**The Faculty of Information Technology**

**FINAL EXAM**

**SEMESTER 2, Year 2022 - 2023**

Class: 72ITSE30303

Subject: Data Structures and Algorithms

Time allowed: 75 minutes

Date taken: April 7th, 2023

Exam format: The exam is entirely in essay format

*Instructions:*

* The exam is divided into three parts.
* Students are not allowed to use any materials during the exam, including books, internet access, cell phones.
* The exam must be completed individually, and students cannot receive any assistance from others.
* For question that require pseudocode, students may use combination of English and programming notation to describe your solution. Your solution should be clear and detailed enough that a programmer could easily implement it.
* Students must submit your completed exam before the end of exam time, Submitting the exam more than 3 minutes after the end time will result in a penalty of 5 points.

**Part A. (1 marks) Answer the following Multiple Choice Questions.**

In……………, search start at the beginning of the list and check every element in the list.

1. Linear search.
2. Binary search.
3. Hash Search.
4. Binary Tree search.

ANSWER: A

Which if the following is/are the levels of implementation of data structure.

1. All of the above.
2. Abstract level.
3. Application level.
4. Implementation level.

ANSWER: A

Stack is also called as.

1. Last in first out.
2. First in last out.
3. Last in last out.
4. First in first out.

ANSWER: A

Queue is also called as.

1. First in first out.
2. Last in first out.
3. First in last out.
4. Last in last out.

ANSWER: A

For a binary search algorithm to work, it is necessary that the array (list) must be.

1. Sorted.
2. Unsorted.
3. In a heap.
4. Popped out of the stack.

ANSWER: A

Following is true about linked list over array:

1. All of the above.
2. Arrays have better cache locality that can make them better in terms of performance.
3. It is easy to insert and delete elements in Linked List.
4. Random access is not allowed in a typical implementation of Linked Lists.

ANSWER: A

After each iteration in bubble sort.

1. at least one element is at its sorted position.
2. one less comparison is made in the next iteration.
3. Both A & B are true.
4. Neither A nor B is true.

ANSWER: A

Which of the following algorithms cannot be designed without recursion.

1. None of the above.
2. Tower of Hanoi.
3. Fibonacci Series.
4. Tree Traversal.

ANSWER: A

An Array is what kind of data structure.

1. Linear.
2. Non- Linear.
3. Complex.
4. All the above.

ANSWER: A

With every push in the stack the top.

1. Increments by one.
2. Decrements by one.
3. Stays there itself.
4. It is always a 0.

ANSWER: A

**Part B. (3 marks) Sorting and Analysis of Algorithms.**

Illustrate the operation of bubble sort by completing the right table below. In successive rows of the table, show the array contents after each pass of the algorithm.

Consider the following array:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 4 | 6 | 2 | 1 |
| 4 | 6 | 2 | 1 | 7 |
| 4 | 2 | 1 | 6 | 7 |
| 2 | 1 | 4 | 6 | 7 |
| 1 | 2 | 4 | 6 | 7 |

* How does Bubble Sort Work?
* Write a Pseudocode of the Bubble Sort algorithm

|  |
| --- |
| begin BubbleSort(list)  **for** all elements of list  **if** list[i] > list[i+1]  swap(list[i], list[i+1])  end **if**  end **for**  **return** list  end BubbleSort |

* Write The Complexity of the Bubble Sort Algorithm: Worst Case, Best Case, Average Case.

The average time complexity of Bubble Sort case is: *O(n²)*

The worst-case time complexity of Bubble Sort is: *O(n²)*

The best-case time complexity of Bubble Sort is: *O(n)*

**Part C. (6 marks) Abstract Data Types**

Use a one-dimensional array to store a class with N students. Know that each student includes the following information: Name (string), Student ID (string) and GPA (float). Write a function that does the following:

* Print the list of students on the screen
* List the students with the highest GPA in the class.
* Show the number of students with an average score >= 5. If not, the message is not available.
* Find a student named X in the classroom (X entered from the keyboard)

|  |
| --- |
| **package** com.minhtan;  **import** java.util.Scanner;  **public** **class** Student {  String name;  String studentId;  **double** GPA;  Scanner in = **new** Scanner(System.***in***);  **public** Student() {  }  **public** **void** input() {  System.***out***.println("Enter student name: ");  **this**.name = in.nextLine();  System.***out***.println("Enter student id: ");  **this**.studentId = in.nextLine();  System.***out***.println("Enter GPA: ");  **this**.GPA = Double.*parseDouble*(in.nextLine());  }  **public** **void** output() {  System.***out***.println("Name: "+**this**.name);  System.***out***.println("Id: "+**this**.studentId);  System.***out***.println("GPA: "+**this**.GPA);  }  **public** **static** **void** main(String[] args) {  Student x = **new** Student();  x.input();  x.output();  }  } |

Class: StudentList

|  |
| --- |
| **package** com.minhtan;  **import** java.util.Scanner;  **public** **class** StudentList {  Student[] students = **null**;  **int** n;  Scanner in = **new** Scanner(System.***in***);  **public** StudentList() {  }  **public** **void** inputStudent() {  System.***out***.println("Enter the amount of students: ");  n = Integer.*parseInt*(in.nextLine());  students = **new** Student[n];  **for**(**int** i = 0; i < n; i++) {  students[i] = **new** Student();  System.***out***.println("The student "+(i+1)+": ");  students[i].input();  System.***out***.println();  }  }  **public** **void** outputStudent() {  System.***out***.println("\n\t-----SHOW LIST OF STUDENTS-----\n");  **for**(**int** i = 0; i < n; i++) {  System.***out***.println("The student "+(i+1)+": ");  students[i].output();  System.***out***.println();  }  }  } |

Class: Main

|  |
| --- |
| **package** com.minhtan;  **public** **class** Main {  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  StudentList list = **new** StudentList();  list.inputStudent();  list.outputStudent();  }  } |

Good Luck!

—————————