

Java institute for Advanced Technology

Department of Examinations



| | |
|-----------------------|---|
| Course – (Leading To) | IIC University BEng (Hons) Software Engineering |
| Unit Name | Data Structures and Algorithms |
| Unit Id | JIAT/DSA |
| Assignment Id | JIAT/DSA /EX/01 |
| Assignment Summary | |
| Duration | 2 Week |
| Submission Via | Online (Student Portal) |
| Document Format | Document Format (Pdf) |

GUIDELINES FOR CANDIDATES

- **Answer only 1 question from Data Structures and 1 question from Algorithms**
- Your studies will be governed by the Java Institute Academic Regulations on Assessment, Progression and Awards.
- Students are expected to use reference books, the Internet, journals and other similar sources in order to accomplish the task specified above.
- Students are expected to refrain from repeating any content in their research document.
- At the re-assessment attempt, the mark is capped and the maximum mark that can be achieved is 40%.

CHEATING AND PLAGIARISM

Both cheating and plagiarism are totally unacceptable and the Institute maintains a strict policy against them. It is YOUR responsibility to be aware of this policy and to act accordingly.

The basic principles are:

- Don't pass off anyone else's work as your own, including coding examples. This is plagiarism and is viewed extremely seriously by the Institute.
- Don't submit a piece of work in whole or in part that has already been submitted for assessment elsewhere. This is called duplication and, like plagiarism, is viewed extremely seriously by the Institute.
- Always acknowledge all of the sources that you have used in your assignment or project.
- If you are using the exact words of another person, always put them in quotation marks.
- Check that you know whether the assignment is to be produced individually or whether you can work with others.
- If you are doing group work, be sure about what you are supposed to do on your own.
- Never make up or falsify data to prove your point.
- Never allow others to copy your work.
- Never lend disks, memory sticks, or copies of your coursework to any other student. in the Institute; this may lead to you being accused of collusion.

Data Structures

Patient's Priority

The hospital's emergency department sees a large volume of patients with various medical conditions. Each patient is assigned a priority level based on the severity of their condition, ranging from 1 (most severe) to 5 (least severe). Patients are treated on a first-come, first-served basis, but if a patient with a higher priority arrives, they should be given immediate care.

The hospital wants to keep track of the patients in the emergency department in real-time and be able to quickly find the patient with the highest priority. They also want to be able to add new patients to the system and update the priorities of existing patients as needed.

What data structure would be appropriate for this scenario, and how can it be used to efficiently find the patient with the highest priority?

Patient 1: Priority level 3, arrived at 9:00 AM

Patient 2: Priority level 1, arrived at 9:15 AM

Patient 3: Priority level 2, arrived at 9:30 AM

Patient 4: Priority level 4, arrived at 10:00 AM

Patient 5: Priority level 1, arrived at 10:30 AM

Assuming that the data structure is correctly implemented, the patient with the highest priority should be Patient 2, who has priority level 1.

(Do Not Use `java.util.*` package)

Student work

Suppose you are working on a project that requires you to maintain a collection of student records. Each record contains a student's name and their corresponding grade on a recent test. You decide to use an appropriate algorithm to store and organize these records.

As you start to add records to the tree, you notice that the tree is becoming unbalanced, with most of the records being on one side of the tree. How would you rebalance the tree to ensure that search and retrieval of records are efficient? Describe the steps you would take to perform the rebalancing.

Assume that you have the following student records:

John: 90
Lisa: 85
Eric: 93
Sarah: 78
David: 85
Alice: 89
Peter: 95
Kelly: 88
Sam: 87
Michelle: 92

(Do Not Use `java.util.*` package)

Data Compression

You are working on a data compression project where you need to compress a large amount of text data. After analyzing the data, you have determined that there are only six distinct characters used in the text: 'a', 'b', 'c', 'd', 'e', and 'f'. You have also determined the frequency of each character as follows:

Character Frequency

a 10

b 4

c 6

d 12

e 20

f 3

You need to compress this data and represent each character with a binary code. How would you go about creating a mapping of each character to its corresponding binary code using a binary tree? Show the steps you took to obtain the binary code for the character 'a'.

(Do Not Use `java.util.*` package)

Algorithms

Managing and Retrieving data efficiently

You are developing a music streaming application that keeps track of the popularity of songs based on the number of times they have been played. Each song is identified by its title, and you want to use a data structure to efficiently manage and retrieve the most popular songs using the Splay Tree algorithm.

Question:

Given the following dataset of songs and their play counts:

1. Song: *"Shape of You"* - Play Count: 500
2. Song: *"Bohemian Rhapsody"* - Play Count: 350
3. Song: *"Rolling in the Deep"* - Play Count: 450
4. Song: *"Hey Jude"* - Play Count: 400
5. Song: *"Smells Like Teen Spirit"* - Play Count: 300

a) Apply the Splay Tree algorithm to construct the Splay Tree based on the play counts of the songs. Show the resulting tree structure after each insertion.

b) After constructing the Splay Tree, perform a search operation to retrieve the song with the highest play count. Show the steps you take to splay the tree and obtain the most popular song.

(Do Not Use `java.util.*` package)

Finding the position of a number

You are developing a program that analyzes the Fibonacci sequence and requires finding a specific number in the sequence. The Fibonacci sequence is a series of numbers in which each number is the sum of the two preceding ones, starting from 0 and 1. You decide to implement a Java program using iteration to find the position of a given number in the Fibonacci sequence.

Question:

Write a Java program that takes an integer input 'target' representing a number to be searched in the Fibonacci sequence. Implement an iterative function to find the position (index) of the given number in the Fibonacci sequence. If the number is not found in the sequence, return -1. Test your program with a target value of 55 and display the output.

For example, given the Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Find the position (index) of the number 55 in the Fibonacci sequence.

Tower of Hanoi algorithm for solving the Tower of Hanoi puzzle

You are working on a robotic arm project that involves manipulating objects with multiple pegs. To test the capabilities of the robotic arm, you decide to implement the Tower of Hanoi puzzle using the arm's movements. The Tower of Hanoi puzzle consists of three pegs and a set of disks of different sizes, initially stacked in increasing order of size on one peg. The goal is to move all the disks to another peg, following the rules that only one disk can be moved at a time, and a larger disk cannot be placed on top of a smaller disk.

Question:

Given a Tower of Hanoi puzzle with three disks initially stacked on the first peg, implement the Tower of Hanoi algorithm to move the disks to the third peg. Show the step-by-step process of moving the disks, indicating the source peg, destination peg, and any auxiliary peg used.

For example, given the initial state of the Tower of Hanoi puzzle with three disks:

- Peg 1: [3, 2, 1]
- Peg 2: []
- Peg 3: []

Perform the Tower of Hanoi algorithm to move the disks to the third peg, following the rules of the puzzle.

The shortest path between two locations on a transportation network

Scenario:

You are working on a ride-sharing application that requires finding the shortest path between two locations on a map. The map consists of multiple interconnected nodes representing different locations, and each node is associated with a specific distance. To calculate the shortest path efficiently, you decide to implement Dijkstra's algorithm.

Question:

Given a map with interconnected nodes and their corresponding distances, implement an appropriate algorithm to find the shortest path between two specified locations. Show the step-by-step process of finding the shortest path, indicating the current node, the calculated distances, and the chosen path at each step.

For example, given the following map:

Nodes: A, B, C, D, E, F

Distances:

A to B: 5

A to C: 2

B to D: 3

B to E: 4

C to D: 1

C to E: 6

D to F: 2

E to F: 4

Find the shortest path from node A to node F using appropriate algorithm, considering the given distances and map structure.

