ALLAMAIQBALOPENUNIVERSITY, ISLAMABAD

**(Department of Computer Science)**

|  |
| --- |
| **WARNING**   1. **PLAGIARISM OR HIRING OF GHOST WRITER(S) FOR SOLVING THE ASSIGNMENT(S) WILL DEBAR THE STUDENT FROM AWARD OF DEGREE/CERTIFICATE, IF FOUND AT ANY STAGE.** 2. **SUBMITTING ASSIGNMENTS BORROWED OR STOLEN FROM OTHER(S) AS ONE’S OWN WILL BE PENALIZED AS DEFINED IN “AIOU PLAGIARISM POLICY”.** |

**Course: Data Structure & Applications (4507)**

**Level: PGD (CS) Semester: Autumn 2024**

**Total Marks: 100 Pass Marks: 50**

**ASSIGNMENT-1**

**(Unit 1 –4)**

**Note: All questions are compulsory. Each question carries equal marks.**

|  |  |
| --- | --- |
| **Q.NO.1** | Define **data structures** and explain their significance in computer science. Discuss the differences between **linear** and **non-linear data structures**, providing examples of each.  How do **abstract data types (ADTs)** fit into this concept, and why are they important in algorithm design and problem-solving? |
| **Q.NO.2** | Explain the concepts of **space complexity** and **time complexity** in relation to data structures. Why are these measures crucial when selecting an appropriate data structure for a specific problem?  Illustrate your explanation with an example of how time complexity impacts the performance of insertion and deletion operations in **data structures** |
| **Q.NO.3** | Define an **array**. How does it differ from other data structures like lists and linked lists?  Provide examples of **1-dimensional arrays** and **multi-dimensional arrays** with suitable examples. |
| **Q.NO.4** | What are some real-world applications of arrays?  Given an array A of size 10, explain how the array stores elements in contiguous memory locations. If the base address of A[0] is 1000 and each element occupies 4 bytes, calculate the memory address of A[5]. |
| **Q.NO.5** | Define a **stack**. How does it operate compared to other data structures like queues or arrays?  Consider a stack with a maximum size of 5. If the stack initially contains the elements 1, 2, 3, explain the process of pushing the elements 4 and 5 onto the stack and the result of popping two elements from the stack. Show the contents of the stack after each operation with the help of figure. Also explain the complexity of the algorithm |

**Course: Data Structure & Applications (4507)**

**Level: PGD (CS) Semester: Autumn 2024**

**Total Marks: 100 Pass Marks: 50**

**ASSIGNMENT-2**

**(Unit 5 –8)**

**Note: All questions are compulsory. Each question carries equal marks.**

|  |  |
| --- | --- |
| **Q.NO.1** | Consider a singly linked list with three nodes storing the values 5, 10, and 15. Draw a diagram representing this list and indicate the address of each node.  Explain the time complexity for accessing an element in a linked list. Compare this with the time complexity of accessing an element in an array. |
| **Q.NO.2** | Explain the concept of **FIFO (First In, First Out)**. Provide a real-world example where a queue would be the most appropriate data structure to use.  Consider a circular queue of size 5 represented by an array A[5]. If the front pointer is at index 1 and the rear pointer is at index 3, show how the enqueue operation works for two more elements, and explain what happens when the rear pointer wraps around the array. |
| **Q.NO.3** | Can binary search be applied to data types other than integers, such as **strings** or **floating-point numbers**? Explain how binary search would work for a sorted array of strings.  Given an array of floating-point numbers {1.1, 2.3, 3.4, 4.6, 5.8}, how would binary search find the value 3.4? List the steps and the intermediate values of the pointers. |
| **Q.NO.4** | Given an array {9, 5, 1, 4, 3}, explain the step-by-step process of sorting it using insertion sort. Show the intermediate steps as each element is placed in its correct position.  Write the **pseudocode** for selection sort and explain the role of the inner and outer loops in the algorithm. |
| **Q.NO.5** | Describe the three main types of **tree traversal** methods: **pre-order**, **in-order**, and **post-order**. Provide an example of when each traversal method might be used.  Describe the two common ways to represent a graph: **adjacency matrix** and **adjacency list**. How does the choice of graph representation affect the performance of graph algorithms? Provide examples of scenarios where one representation is preferred over the other. |

**Note :** Students shall also perform programming exercises as given in the handouts.