# Assignment 6: Medians and Order Statistics & Elementary Data Structures

## Overview:

This assignment is designed to deepen your understanding of medians, order statistics, and elementary data structures. You will implement algorithms related to these topics, analyze their performance, and discuss their practical applications. The assignment is divided into two parts, each with specific deliverables. By the end of this assignment, you will have a solid grasp of both the theoretical and practical aspects of these essential algorithmic concepts.

## Part 1: Implementation and Analysis of Selection Algorithms

### **Objective:**

Implement algorithms for selecting the \(k^{th}\) smallest element (order statistics) in an array. This includes the deterministic algorithm for selection in worst-case linear time and a randomized algorithm for selection in expected linear time.

### **Instructions:**

1. \*\*Implementation:\*\*  
- Implement the \*\*deterministic algorithm\*\* for selection in worst-case linear time (e.g., Median of Medians).  
- Implement a \*\*randomized algorithm\*\* for selection in expected linear time (e.g., Randomized Quickselect).  
- Ensure your implementation is efficient and handles edge cases such as arrays with duplicate elements.

2. \*\*Performance Analysis:\*\*  
- Provide a detailed analysis of the time complexity for both the deterministic and randomized selection algorithms.  
- Explain why the deterministic algorithm achieves \(O(n)\) time complexity in the worst case, while the randomized algorithm achieves \(O(n)\) time complexity in expectation.  
- Discuss the space complexity and any additional overheads associated with both algorithms.

3. \*\*Empirical Analysis:\*\*  
- Empirically compare the running time of the deterministic and randomized selection algorithms on different input sizes and distributions (e.g., random, sorted, reverse-sorted).  
- Discuss the observed results and relate them to your theoretical analysis.

## Deliverables:

- Well-documented source code for both the deterministic and randomized selection algorithms.  
- A detailed report discussing your implementation, performance analysis, and empirical results.  
- Clear explanations of the time complexity analysis for both algorithms.

## Part 2: Elementary Data Structures Implementation and Discussion

### **Objective:**

Explore and implement basic data structures, including arrays, stacks, queues, and linked lists. Analyze their performance and discuss their practical applications.

### **Instructions:**

1. \*\*Implementation:\*\*  
- Implement the following data structures from scratch using Python:  
 - \*\*Arrays and Matrices:\*\* Implement basic operations such as insertion, deletion, and access.  
 - \*\*Stacks and Queues:\*\* Implement both stack and queue operations using arrays.  
 - \*\*Linked Lists:\*\* Implement singly linked lists with operations such as insertion, deletion, and traversal.  
 - Optionally, implement \*\*rooted trees\*\* using linked lists to represent the nodes and their relationships.

2. \*\*Performance Analysis:\*\*  
- Analyze the time complexity for the basic operations of each data structure.  
- Discuss the trade-offs between using arrays versus linked lists for implementing stacks and queues.  
- Compare the efficiency of different data structures in specific scenarios.

3. \*\*Discussion:\*\*  
- Provide a discussion on the practical applications of these data structures in real-world scenarios.  
- Highlight scenarios where one data structure may be preferred over another due to factors like memory usage, speed, and ease of implementation.

## Deliverables:

- Well-documented source code for the implementations of arrays, stacks, queues, and linked lists.  
- A report discussing the performance analysis and practical applications of the data structures.  
- Clear explanations of the time complexity analysis for the operations of each data structure.

## Submission Instructions:

1. \*\*GitHub Repository:\*\*  
- Create a new GitHub repository for this assignment.  
- Upload the following materials to your repository:  
 - Your Python implementations for both parts of the assignment.  
 - A report detailing your design choices, implementation details, and analysis.  
 - A README file with instructions on how to run your code and a summary of your findings.

2. \*\*Submit the GitHub Repository Link:\*\*  
- Submit the link to your GitHub repository as your final submission.