Data Structures and Algorithms

Mohammed Fahad

July 2, 2025

Syllabus

Basic Concepts of Data Structures

Definitions; Data Abstraction; Performance Analysis - Time & Space Complexity, Asymptotic Notations; Polynomial representation using Arrays, Sparse matrix (Tuple representation); Stacks and Queues - Stacks, Multi-Stacks, Queues, Circular Queues, Double Ended Queues; Evaluation of Expressions- Infix to Postfix, Evaluating Postfix Expressions.

1 Definitions

- **Data Structures:** ways of organizing and storing data in a computer so that it can be accessed and modified efficiently. Types:
  1. Linear: Arrays, Linked Lists, Stacks, Queues
  2. Non-linear: Trees, Graphs
- **Data Abstraction:** concept of hiding the internal details of how data is stored or maintained and only showing the essential features or operations that can be performed on the data.

2 Performance Analysis

1. **Time Complexity**
2. **Space Complexity**

3 Stack

It follow FILO (First In Last Out) scheme

- pop - Removes from top
- push - Adds to top
- peek/top - See topmost element

4 Queue

It follow FIFO (First In First Out) scheme

- pop - Removes from front
- push - Adds to rear
- peek/top - See frontmost element

5 Addition of sparse polynomial

All the polynomials are stored inside an array of structures:

```c
// Structure to represent a term
typedef struct {
    int coeff;
    int expo;
} Term;

Term polynomial[] = {{2, 3},{4, 0}} // 2x^3 + 4
```

Listing 1: Sparse Polynomial Addition Outline

5.1 Logic when adding 2 polynomials

let it be poly1 (with i as indexing), poly2 (with j as indexing) & result (with k as indexing)

- If poly1[i].exp == poly2[j].exp: add coefficients
- If poly1[i].exp is greater than poly2[j].exp: Copy over poly1[i]
- If poly1[i].exp is less than poly2[j].exp: Copy over poly2[j]