DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
KATHMANDU UNIVERSITY

Subject: Data Structures and Algorithms — Course Code: AICS 101
Credit: 3 — F.M: 100
Type: Core [Theory + Practical (Lab)]

Course Description

This course includes the basic foundations of data structures and algorithms. It covers concepts of various data structures like stack, queue, list, tree and graph. This course also includes the idea of sorting and searching.

Objectives

- To introduce data abstraction and data representation in memory
- To describe, design and use of elementary data structures such as stack, queue, linked list, tree and graph
- To discuss decomposition of complex programming problems into manageable sub-problems
- To introduce various standard algorithms and their complexity

Prerequisites

A good knowledge of any programming language such as C, C++ or Java. Fundamental mathematical knowledge is necessary.

Contents

1. Fundamentals [1 hour]
   - 1.1. Data types, data structures and abstract data types (ADT)
   - 1.2. Fundamental data structures
     - 1.2.1. Arrays
     - 1.2.2. Structures
   - 1.3. Recursion

2. Algorithm Analysis [3 hours]
   - 2.1. Algorithm Specification
   - 2.2. Performance analysis
     - 2.2.1. Space complexity
     - 2.2.2. Time complexity
     - 2.2.3. Asymptotic notation - Big O Notation, Omega Notation, Theta Notation

3. Stack and Queue [7 hours]
   - 3.1. The Stack ADT - Definition, and operations
   - 3.2. Array-based stack implementation
   - 3.3. Applications of Stack
     - 3.3.1. Reversing a string
     - 3.3.2. Evaluation of expressions

3.3.2.1. Expressions: Prefix, infix, postfix
3.3.2.2. Infix to Postfix Conversion
3.3.2.3. Evaluating Postfix Expressions
3.4. The Queue ADT - Definition, and Operations
3.5. Array-based queue implementation
3.6. Circular queue
3.7. Priority queue
3.7.1. Ascending priority queue
3.7.2. Descending priority queue

4. Linked List [6 hours]
4.1. Singly linked list
4.1.1. Definition
4.1.2. Operations
4.1.2.1. Insertion
4.1.2.2. Deletion
4.1.2.3. List traversal
4.1.3. Implementation of stack and queue using a singly linked list
4.2. Circular Linked List
4.3. Doubly Linked List

5. Trees [12 hours]
5.1. Introduction
5.1.1. Concept and definition
5.1.2. Terminologies: Root node, leaf node, internal node, parent, child, sibling, degree of a tree, height of a tree, depth of a tree, balance factor
5.1.3. Representation of Trees
5.2. Binary Tree
5.2.1. Definition and properties
5.2.2. Types of binary trees - Skewed binary tree, Strictly binary tree, Complete binary tree, Almost complete binary tree
5.2.3. Basic operations
5.2.4. Binary tree traversal - Pre-order, in-order, post-order
5.3. Heaps
5.3.1. Definition and properties
5.3.2. Operations - insertion, deletion, increase-key, decrease-key
5.4. Search Trees
5.4.1. Binary Search Tree (BST)
5.4.1.1. Definition
5.4.1.2. Operations - Search, insertion and deletion
5.4.2. Balanced Search Tree - AVL Tree
5.4.3. M-way Search Tree - B-Tree, B+-Tree

6. Graphs [8 hours]
6.1. Introduction
6.1.1. Concept and definition
6.1.2. Terminologies:
6.2. Types of graphs
6.3. Graph representation

6.3.1. Adjacency Matrix
6.3.2. Incidence Matrix
6.3.3. Adjacency List
6.4. Operations on Graph
6.5. Graph Traversal - Depth-First Search (DFS), and Breadth-First Search (BFS)
6.6. Minimum Spanning Tree
6.6.1. Kruskal's Algorithm
6.6.2. Prim's Algorithm
6.7. Shortest-path algorithm - Dijkstra's algorithm

7. Sorting [7 hours]
7.1. Selection Sort
7.2. Insertion Sort
7.3. Quick Sort
7.4. Merge Sort
7.5. Heap Sort
7.6. Shell Sort

8. Searching and Hashing [4 hours]
8.1. Basic Search Techniques
8.1.1. Sequential Search
8.1.2. Binary Search
8.2. Hashing
8.2.1. Introduction to Hashing
8.2.2. Hash function and hash tables
8.2.3. Collision resolution techniques

Textbooks

1. Langsam, Y., Augenstein, M., & Tenenbaum, A. M. (2006). *Data Structures using C and C++* (2nd ed.). New Jersey: Prentice Hall.
2. Horowitz, E., Sahni, S., & Anderson-Freed, S. (2008). *Fundamentals of Data Structures* (2nd ed.). University Press.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). MIT press.
4. Sedgewick R., & Wayne K. (2011). *Algorithms* (4th ed.). Addison-Wesley Professional.
5. Rowe, G. W. *Introduction to Data Structure and Algorithms with C and C++*. Prentice Hall.
6. Kruse, R. L., Leung, B. P., Tondo, C. L., Mogalla, S. (2006). *Data Structure and Program design in C* (2nd ed.). Pearson India.