AE2ADS: Algorithms Data Structures and Efficiency

Lecturer: Heshan Du
Email: Heshan.Du@nottingham.edu.cn

University of Nottingham Ningbo China

Topics Covered by AE2ADS

Mathematical Tools for Algorithm Analysis

1. Algorithm Efficiency Analysis
    - Big-Oh
    - Big-Omega, Big-Theta and Little-Oh
2. Recurrence and Master Theorem

Topics Covered by AE2ADS

Algorithms and Data Structures

1. Stacks and Queues
2. List Abstractions
3. Tree Structures
4. Sorting Algorithms
5. Priority Queues and Heaps
6. Maps and Hash Tables
7. Sorted Maps and Binary Search Trees
8. Graphs and Graph Algorithms

Algorithm Efficiency Analysis

- Algorithm Efficiency Analysis
  - Experimental studies
  - Theoretical analysis
- Pseudocode
- Big-Oh Notation
- Relatives of Big-Oh
  - Big-Omega
  - Big-Theta
  - Little-Oh

Abstract Data Types (ADTs)

- An abstract data type (ADT) is an abstraction of a data structure
- An ADT specifies:
  - Data stored
  - Operations on the data
  - Error conditions associated with operations

Recurrences and Master Theorem

- What is a recurrence?
- Methods for solving recurrences
  - Substitution method
  - Recursion-tree method
  - Master method

Linked Lists, Stack and Queue

- Singly linked lists
- Doubly linked lists
- Stack ADT
  - last-in first-out
  - array-based stack
- Queue ADT
  - first-in first-out
  - array-based queue

List and Positional List

- List ADT
- Array Lists
- Growable array-based array list
  - Incremental strategy
  - Doubling strategy
  - Amortized time analysis
- Positional List ADT

Tree and Binary Tree

- Tree definition and ADT
- Tree terminology, depth, height
- Tree traversal algorithms
- Binary tree definition and ADT
- Binary tree traversal algorithms
  - Preorder, Postorder and Inorder Traversal
  - Euler Tour Traversal

Abstract Data Types vs. Data Structures

Abstract Data Type (ADT)	Data Structure
Stack	Array
Queue	Singly Linked List
List	Doubly Linked List
Positional List	Linked Structure
Tree	
Binary Tree	

Simple Sorting Algorithms

- Bubble Sort
- Selection Sort
- Insertion Sort

Merge Sort and Quick Sort

- Divide and conquer
- Merge Sort
  - Merging two sorted sequences
  - Complexity analysis
- Quick Sort
  - Partition
  - Worst-case running time
  - Expected running time
- In-Place Quick Sort

Priority Queues and Heaps

- Priority queue ADT
- Sequence-based priority queue
- Priority queue sort
- What is a heap?
- Basic operations and their complexity
- Heap-sort

Maps and Hash Tables

- Map ADT
- A list-based map
- Hash tables
- Collision handling
  - Separate chaining
  - Linear probing
  - Double hashing

Sorted Map ADT and Binary Search Trees

- Sorted Map ADT
- What is a binary search tree?
- Basic operations and their complexity

Graph and Graph Algorithms

- Definition of a graph and its related terminology
- Graph traversal algorithms and their complexity
  - Breadth-first search
  - Depth-first search
- Topological sort
- Minimal spanning tree
- Dijkstra's shortest path algorithm and its complexity

Abstract Data Types vs. Data Structures

Abstract Data Type (ADT)	Data Structure
Priority Queue	Heap
Map	Hash table
Sorted Map	Binary Search Tree
Graph	Adjacency Matrix
	Adjacency List

Assessment

- Coursework 25%
- Final exam 75%

Format of the Final Exam

- Answer all FOUR questions.
- The total mark is 100.
- No calculators are permitted in this examination.

Be prepared!

Good luck !