CS2040 NOTES

Special Term Part II AY18/19

Hanming Zhu

Overview

Complexities

1. Analysing Algorithm
    a. What is an Algorithm?
    b. Analysis of Algorithms
    c. Big-O Notation
    d. Worst Case, Best Case, Average Case
2. Mathematical Equations

Searching

1. Sequential Search
2. Binary Search
3. Hashing

Sorting

Comparison Based and Iterative

1. Selection Sort
2. Insertion Sort
3. Bubble Sort without flag
4. Bubble Sort with flag

Comparison Based and Recursive

5. Merge Sort
6. Quick Sort

Non-Comparison Based

7. Radix Sort

Using Advanced Data Structures

8. Heap Sort
    a. Not cache friendly
9. Topological Sort

Physical Data Structures

1. Arrays
    a. Circular Arrays
2. Linked Lists
    a. Basic Linked List
    b. Tailed Linked List
    c. Circular Linked List
    d. Doubly Linked List

Logical Data Structures

Linear

1. Lists
    a. Arrays
    b. Linked Lists

2. Stacks
    a. Arrays
    b. Linked Lists
3. Queues
    a. Circular Arrays
    b. Tailed Linked List
    c. Deque
        i. Circular Arrays
        ii. Tailed Linked List

Depends on Implementation

4. HashMap
    a. Arrays
    b. Collisions and Resolutions
        i. Chaining
            1. Array of Linked Lists
            2. Decreasing Efficiency
                a. Table Rehashing
        ii. Linear Probing
            1. Deletion
                a. Lazy Deletion
            2. Primary Clustering
                a. Modified Linear Probing
        iii. Quadratic Probing
            1. Theorem of Quadratic Probing
            2. Secondary Clustering
        iv. Double Hashing
            1. Cannot resolve to 0
            2. Analysis

Non-Linear

5. Priority Queue *(can be linear but we are learning non-linear)*
    a. Binary Heap *(logic is non-linear)*
        i. Array *(implemented "linearly")*
6. Set
    a. Hash Table
        i. Array (not elaborated)
    b. Union-Find Disjoint Sets
        i. Array
            1. Parent
            2. Rank
        ii. Inverse Ackermann Function
        iii. Static Data Structure
7. Ordered Map
    a. Array
        i. Sorted
        ii. Unsorted
    b. BST

        i. Array (Similar to Binary Heap)
        ii. Array (Similar to UFDS)
        iii. Nodes (Linked List)
    c. AVL Trees
        i. Balancing the Tree
        ii. Proof: Bounds of h
8. Graphs
    a. Terminologies
    b. Adjacency Matrix
        i. 2D Array
    c. Adjacency List
        i. Array of Linked Lists
        ii. Array of Arrays
    d. Edge List
        i. Array
9. Graph Operations and Analysis of Implementations
    a. Supporting Operations
    b. BFS
        i. Queue and Arrays
    c. DFS
        i. Recursion/Stack and Array
    d. Applications of Graph Traversal
        i. Topological Sort
            1. Proof: Every DAG has a Topological Ordering
            2. Kahn's Algorithm
                a. BFS
            3. DFS
10. Minimum Spanning Tree
    a. Understanding Real World Application
    b. Prim's Algorithm
        i. PriorityQueue and Arrays
        ii. Proof: Prim's Algorithm works
    c. Kruskal's Algorithm
        i. UFDS and Edge List
        ii. Proof: Kruskal's Algorithm works
11. Single-Source Shortest Path
    a. Supporting Data Structure
        i. Array
    b. BFS for Unweighted
    c. Generic SSSP Algorithm
    d. Bellman Ford's Algorithm for All General Graphs
        i. Adjacency List or Edge List
    e. One-pass Bellman Ford's Algorithm
    f. BFS/DFS for Trees
    g. Original Dijkstra's Algorithm
        i. PriorityQueue and HashTable
            1. Binary Min Heap
        ii. bBST

Cheat Sheet