

# CS251 Comprehensive Review

Theo Park

## Contents

<b>1 Runtime Analysis</b>	<b>4</b>
1.1 Power Law . . . . .	4
1.2 Runtime Expressions . . . . .	4
1.3 Asymptotic Runtime Analysis . . . . .	4
1.4 Recursive Relationship . . . . .	4
<b>2 Intro to Data Structures</b>	<b>4</b>
2.1 Array . . . . .	4
2.2 Linked List . . . . .	4
2.3 Stack . . . . .	4
2.4 Queue . . . . .	4
2.5 Binary Heap . . . . .	4
2.6 Tree . . . . .	4
<b>3 Sorting Algorithms</b>	<b>4</b>
3.1 Bubble Sort . . . . .	4
3.2 Selection Sort . . . . .	4
3.3 Insertion Sort . . . . .	4
3.4 Shell Sort . . . . .	4
3.5 Heap Sort . . . . .	4
3.5.1 Building a Heap – Top-down v.s. Bottom-up . . . . .	4
3.5.2 Sort Down Algorithm . . . . .	4
3.6 Merge Sort . . . . .	4
3.6.1 Merge Algorithm . . . . .	4
3.7 Quick Sort . . . . .	4
3.7.1 Pivot and Partition . . . . .	4
3.8 Decision Tree and $\Omega(n \log n)$ Limit for Comparison Sorting Algorithms . . . . .	4
3.9 Counting Sort . . . . .	4
3.10 Radix Sort . . . . .	4
<b>4 Hash Tables</b>	<b>4</b>
4.1 Division Method . . . . .	4
4.2 Multiplication Method . . . . .	4
4.3 Collision . . . . .	4
4.3.1 Chaining . . . . .	4
4.3.2 Open Addressing . . . . .	4
<b>5 Search Tree</b>	<b>4</b>
5.1 Binary Search Tree and Its Limit . . . . .	4
5.2 2-3 Tree . . . . .	4
5.3 Red-Black Tree . . . . .	4
5.4 Left-Leaning Red-Black Tree . . . . .	4
5.4.1 Deletion in LLRBT . . . . .	4

<b>6</b>	<b>Graph Traversal</b>	<b>4</b>
6.1	Adjacency Matrix and List . . . . .	4
6.2	DFS . . . . .	4
6.3	BFS . . . . .	4
<b>7</b>	<b>Directed Graphs</b>	<b>4</b>
7.1	Strong Connectivity . . . . .	4
7.1.1	Brute-force Strong Connectivity Algorithm . . . . .	4
7.1.2	Brute-force using Stack . . . . .	4
7.1.3	Strongly Connected Components and Kosaraju's Algorithm . . . . .	4
7.2	Directed Acyclic Graphs . . . . .	4
7.2.1	Topological Sort . . . . .	4
<b>8</b>	<b>Weighted Graphs</b>	<b>4</b>
8.1	Shortest Path . . . . .	4
8.1.1	Dijkstra's Algorithm . . . . .	4
8.1.2	Bellman-Ford Algorithm . . . . .	4
8.2	Articulation Points . . . . .	4
8.3	Minimum Spanning Tree . . . . .	4
8.3.1	Cycle and Cut Properties . . . . .	4
8.3.2	Prim's Algorithm . . . . .	4
8.4	Union-Find . . . . .	4
8.4.1	Kruskal MST Algorithm . . . . .	4
<b>9</b>	<b>Strings</b>	<b>4</b>
9.1	Brute-force String Pattern Matching . . . . .	4
9.2	KMP Algorithm . . . . .	4
9.3	Trie . . . . .	4
9.4	PATRICIA . . . . .	4
9.5	Huffman Coding . . . . .	4



## 1 Runtime Analysis

### 1.1 Power Law

### 1.2 Runtime Expressions

### 1.3 Asymptotic Runtime Analysis

### 1.4 Recursive Relationship

## 2 Intro to Data Structures

### 2.1 Array

### 2.2 Linked List

### 2.3 Stack

### 2.4 Queue

### 2.5 Binary Heap

### 2.6 Tree

## 3 Sorting Algorithms

### 3.1 Bubble Sort

### 3.2 Selection Sort

### 3.3 Insertion Sort

### 3.4 Shell Sort

### 3.5 Heap Sort

#### 3.5.1 Building a Heap – Top-down v.s. Bottom-up

#### 3.5.2 Sort Down Algorithm

### 3.6 Merge Sort

#### 3.6.1 Merge Algorithm

### 3.7 Quick Sort

#### 3.7.1 Pivot and Partition

### 3.8 Decision Tree and $\Omega(n \log n)$ Limit for Comparison Sorting Algorithms

### 3.9 Counting Sort

### 3.10 Radix Sort

## 4 Hash Tables

### 4.1 Division Method

### 4.2 Multiplication Method

### 4.3 Collision

#### 4.3.1 Chaining

#### 4.3.2 Open Addressing

## 5 Search Tree