CSC 130 Data Structures and Algorithm Analysis – Fall 2016

Final Exam Study Guide

Section 1 Exam Time: Thursday, December 15, 2016 10:15AM – 12:15PM ARC 3004

Section 2 Exam Time: Tuesday, December 13, 2016 3:00PM – 5:00PM RVR 1006

Final exam will be a close book, close note, 2-hour examination. The exam will focus on, but not limited to, the lecture notes after the midterm.

Part I Basics

* Abstract Data Type
* Theoretical Analysis of Algorithms
  + Time and space cost
  + Growth rate of functions
* Asymptotic Notations
  + Big-Oh, Big-Theta, Big-Omega: Definitions and mathematical implications (e.g., bound)
    - Focus on Big-Oh notation
* Mathematical Proof
  + Inductive proof
  + Proof by contradiction

Part II Data Structures

* Arrays
  + Index and address computation
  + Insert, delete, access
* Lists
  + Traversal
  + Singly linked list, doubly linked list
  + Insert, delete, access with different positions and using different lists and pointers
* Stacks
  + Definition
  + Push, pop
    - Practice how these methods work in practical problems/applications
  + Different implementations
* Queues
  + Definition
  + Enqueue, dequeue
  + Different implementations
* Sets, Dictionaries, Hash Tables
  + Definition of sets and dictionaries
  + Know different data structures for implementing dictionaries
    - Runtime
  + Know iterators and comparators
  + Know the idea of hash tables
  + Understand hash functions and collision resolution
    - Separate chaining
    - Linear probing
    - Quadratic probing
    - Double hashing

Part III Trees

* Trees, Binary Trees
  + Definition, properties, terminologies (e.g., parent, child, height, path, etc.)
  + Traverse (pre-order, in-order, post-order, level order)
* Binary Search Trees
  + Definition, search order
  + Insert, delete, search, findMin/Max
  + Performance cost of these operations
* Balanced Trees, AVL Trees
  + Definition of Balance and AVL trees
  + Insertion
    - Left/right rotation, Single/double rotation, 4 cases to apply rotations
    - Spend some time on this
  + Removal
    - Understanding at a high level
  + Running time for search, insertion, and removal
  + Where does insertion and delete happen?
* Splay Trees
  + Splay operation
    - Two easy cases, zig-zig case, and zig-zag case
  + Amortized running time
    - Definition and implication
* B-trees, 2-3 Trees
  + Definition
  + Insertion
    - Two rules and when to use them
  + Removal
    - Understanding at a high level
* Red-Black Trees
  + Definition (I follow the definition of the red book, i.e., left-leaning red-black tree)
    - Symmetric order, perfect black balance
    - Relation with B-trees
  + Insertion
    - Left rotation, right rotation, color flip
    - If you have hard time understanding the rules in the book, I suggest you think about it from a different angle (i.e., LLRB trees have 1-1 correspondence with B-trees)
  + Performance of insertion, removal, and search
* Priority Queues
  + Definition
  + Different implementations
  + Heaps
    - Max-heap order
    - Insertion, removeMax
    - Performance analysis

Part IV Sorting

* Sorting Problems
  + Problem definition, in-place, stable sort, etc.
* Heapsort
  + Performance analysis
* Selection Sort, Insertion Sort
  + Idea, loop invariant,
  + Implementation
  + Performance
* Mergesort, Quick Sort
  + Idea: divide-and-conquer
  + Implementation: partitioning, picking pivot, etc.
  + Performance analysis
* Analysis of comparison sort
  + What is the ceiling of the performance and why?
* Linear sorting
  + Know the concept

Part V Graphs

* Terminology
* Representations: adjacency matrix, adjacency lists
  + Runtime for each operation; space cost
  + Know which one to use in different scenarios/applications
* Traversal: depth-first search, breadth-first search
  + Understand how they would process graphs
  + Runtime
  + Know the appropriate data structure for implementing each strategy
* Topological sort
  + Definition of DAGs and topological ordering
  + Know how to use DFS reverse postorder to implement topological sort
* Shortest path problems
  + Know the variants of shortest path problems
  + Dijkstra’s algorithm
    - Applicability
    - Edge relaxation
    - Understand how it would process graph
    - Understand the implementation and the data structure used
  + Shortest path for acyclic graphs (i.e., using topological sort)
    - Know the idea of the topological sort-based algorithm
  + Shortest path for graphs with negative weights
    - Know the idea of Bellman-Ford algorithm
* Minimum spanning tree
  + Definition of spanning tree
  + Prim’s algorithm
    - Understand how it would process graph
    - Understand the lazy and eager implementations
      * Data structure used
      * Runtime
  + Kruskal’s algorithm
    - Understand how it would process graph
    - Understand the implementation
      * Union-find
      * Data structure used
      * Runtime

Part VI Algorithm Design Techniques and Misc.

* Divide-and-conquer
  + Understand the idea
  + Binary search
    - Understand the implementation
      * Runtime
  + Know how to solve recurrence
    - Iterative substitution
    - Recurrence tree
    - Master theorem
* Greedy algorithms
  + Optimization problems
  + Understand “greedy”
  + Know the exemplary problems: coin change, scheduling problem, Huffman codes
    - Examples
* Dynamic programming
  + Know the idea of DP and differences between DP and divide-and-conquer
  + Know bottom-up and top-down DP
* P and NP
  + Know the definition of P and NP
  + Know what the difference is between an optimization problem and a decision problem
  + Know what is meant by a “polynomial time reduction”
  + Know the definition of NP-complete problem