Review on Final Exam

Ye Yang
Stevens Institute of Technology

General Instruction

- Final exam is focusing on the second half, however, involving application of earlier concepts/standard operations of arrays, lists, stacks, queues, and complexity analysis.
- You can bring a 1-page cheat sheet.
- No coding is required to implement standard data structure operations, but you may be asked to provide Pseudo-code for applying standard operations to implement a non-standard operation.

Recursion

- When to use recursion
- Converting a recursive definition (e.g.: heightOfaBST) to code
- Tracing recursive methods such as undefined foo method. What does it do?
- Complexity of recursive code

Binary Search Trees

- Use of BST's – when to use BST's Standard operations – insert, delete, find, pre/post/in order traversals
- Non-standard operations – findMax, findMin
- Complete, full and perfect trees
- Complexity of BST operations
  - Worst case?

Heaps/Priority Queues

- Max and min heaps
- Standard operations – build heap, deleteMin/Max, insert
- Non-standard operations – findMaxOnMinHeap, deleteSecondSmallest
- Building a heap from a given data set
- Implementation details – using an array, view heap as a complete tree
- Lab 10 discussion

Hash Tables

- Definitions, collisions, linear/quadratic probing, separate chaining, load factors
- Displaying tables given an input dataset and a hash function
- Complexity of hashTable operations
- Lab 11 discussion

Sorting

- Comparison-based sorting – selectionSort, insertionSort
- Advanced sorting – mergeSort, quickSort, and heapSort
- Counting-based sorting – bucket sort
- Complexity of sorting algorithms: average case, worst case
- Given a data set, applying a sorting algorithms with **steps**
  - Understand and be able to express pseudo-code

Graphs

- Definition – nodes, edges, graph connectivity, graph density
- Graph representations – Adjacency Matrix, Adjacency Lists, space complexity
- Applications of graphs
- Traversing a graph using BFS and DFS algorithms
- Tracing Dijkstra's algorithm to calculate single source shortest paths
- No coding of graph algorithms is required

BACKUP: EARLY DATA STRUCTURES

Arrays /ArrayLists

- Use of arrays – what applications are best implemented using arrays
- Standard operations –
  - add(object)
  - remove(object)
  - contains(object)
- Complexity of array algorithms – search (sorted/unsorted), find max, insert (sorted/unsorted), find median, reverse etc..

Linked Lists

- Use of LL's – when to use LL's
- types of LL's – singly, doubly, circular, multilinked
- standard operations: add(object), remove(object), contains(object)
- non-standard operations – findMax/Min, reverse, etc..
- complexity of LL operations

Stacks and Queues

- Standard operations on stacks – pop, push, top, empty
- Standard operations on queues – enqueue, dequeue, empty
- Solving problems using stacks – eg: reversing an array or LL
- Solving problems using queues
- Implementing stacks and queues with arrays or LL's – complexity issues
- Complexity of stack and queue operations