CSE 12 - Winter 2021 Final Exam Review Sheet

CSE 12 topics

The final exam is meant to be comprehensive that may reflect materials covered in PAs, lectures, discussion, quizzes, and reading materials. Everything that we have covered in classes may appear in the final exam.

1. Java generics
   - generic class and function (How to write, how to call)
   - various containers in Java (ArrayList, LinkedList, Stack, Queue, PriorityQueue, HashSet, HashMap, etc)
   - Exception handling in Java
   - ADT vs Backend Data Structure
2. ArrayList
   - Implementation of ArrayList using arrays
   - Basic operation: add, remove, find. Understand the meaning of capacity in ArrayLists
   - The efficiency of basic ArrayList operations
   - circular array for streams
3. LinkedList implementation
   - Definition of Singly linked lists
   - Sentinel nodes and their purposes
   - Basic operations for LinkedLists (add, remove, search) and their efficiencies
4. Hashing
   - Idea of Hashing (use array as backend storage), rehashing and load factor
   - Hash function (basic hashing functions, string to int hash functions)
   - Collision resolution (separate chaining and linear probing)
   - Efficiency of basic hashing (insert, delete, search)

5. Stack and Queue
   - basic queue adaption
   - basic queue operations (enqueue, dequeue, peek)
   - application of queue for BFS
   - basic stack adaption
   - basic stack operation (push, pop, peek)
   - application of stack for DFS

5. Priority queue
   - Complete binary tree and definition of a heap
   - Array-based heap implementation (parent and children positions for both situations where heap elements start at index 0 or 1, percolate down and bubble up operations, heapify an array)
   - Priority queue (push, pop, peek)

7. Tree
   - Definition of trees (node, child, leaf, internal nodes, height, full tree, complete tree)
   - Binary tree
     - i. insert, delete, and search in trees
   - Binary search tree
     - i. definition of a BST
     - ii. Insert, delete, search
     - iii. three order of traversals

8. Runtime analysis
   - Big O, Big Omega, Big Theta formal and intuitive definitions
   - Analysis of a code's runtime in tightest bound
   - Analysis of runtime for data structures we have implemented
   - Amortized analysis

9. Sorting
   - Comparison based sorting (bubble, insertion, selection, merge, quick)
   - Runtime analysis of sorting algorithms (worst case and best case)

10. Streams/Iterators
   - input and output streams
   - implementation of streams and iterators