Exam 1 Outline

The following outlines the topics you should know, and the things you need to be able to do, for the exam. In general, you will not be responsible for C++ code presented in class, except as noted in the outline; however, you may need to read and understand C++ code presented on the exam. The exam will be closed book, closed notes, and closed computer.

Introduction
- What is the point of this class?

Math Review
- Floors, ceilings, exponents and logarithms: Definitions and manipulations
- Factorials and Stirling’s approximation
- Series: Definitions, manipulations, arithmetic and geometric series closed form
- Modular arithmetic
- Proofs: Know definition, components, and how to use the following
  - Proof by induction
  - Proof by counterexample
  - Proof by contradiction
- Recursion
  - Know definition and rules
  - Analyze running time of recursive algorithm

C++ Review
- Know definitions and how to use the following
  - Class, method, encapsulation
  - Constructor, destructor, accessor, mutator
  - Reference variable (\&x) and call by reference
  - Copy constructor, operator overloading, operator=
  - Templates

Algorithm Analysis
- Why analyze an algorithm?
- What do we measure and how do we measure it?
- Line-by-line analysis
- Best-case, worst-case and average-case analysis
- Rate of growth: Definitions and notation (O, \Omega, \Theta, o)
- Maximum subsequence sum problem
  - Definition
  - Four different algorithms
  - Analysis of each algorithm
- Binary search problem: Definition, algorithm, analysis
Abstract Data Types

- Lists
  - Operations: Insert, Delete, Search
  - Implementations: vectors, singly-linked lists, double-linked lists, sentinels
  - Analysis of operations for each implementation

- Stacks
  - Operations: Push, Pop, Top
  - Implementations: linked-list, vector
  - Analysis of operations for each implementation

- Queues
  - Operations: Enqueue, dequeue
  - Implementations: linked-list, vector
  - Analysis of operations for each implementation

- Standard Template Library (STL)
  - Use of vector, list, stack and queue template classes
  - Use of iterators

Trees

- Definitions: root, leaf, child, parent, ancestor, descendant, path, height, depth
- Binary tree: Definition, traversals
- Binary search tree (BST)
  - Definition
  - Operations: Insert, Delete, Search, FindMin, FindMax, traversals
    - Know how to perform these on a BST and show resulting BST
    - Know worst-case and average-case analysis of performance

- AVL trees
  - Definition
  - Operations: Rotations, Insert, Lazy Delete, Search, FindMin, FindMax, traversals
    - Know how to perform these on an AVL tree and show resulting AVL tree
    - Know worst-case performance

- Splay trees
  - Definition
  - Operations: Rotations, Zig-Zag, Zig-Zig, Insert, Delete, Search, FindMin, FindMax, traversals
    - Know how to perform these on a Splay tree and show resulting Splay tree
    - Know amortized cost per operation

- B-trees
  - Definition and properties
  - M and L, and how to choose them
  - Operations: Insert, Delete, Search
    - Know how to perform these on a B-tree and show resulting B-tree
    - Know worst-case performance

- STL set and map classes
  - Differences
  - How to use them