Exam 2 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.

First, here are topics from the first exam you still need to know.

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, counterexample, 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, STL, iterators

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)

Abstract Data Types
- General use of list, stack and queue ADTs

Trees
- Definitions: root, leaf, child, parent, ancestor, descendant, path, height, depth
- Binary tree: Definition, traversals
Here are new topics since the first exam.

Hashing
- Hash functions, choosing good hash functions
- Collision
- Load factor
- Know algorithms and analysis for the following
  - Collision resolution by chaining
  - Collision resolution by open-addressing
    - Linear, random and quadratic probing
    - Double hashing
  - Rehashing

Priority Queues (Heaps)
- Definition
- Operations: insert, deleteMin, decreaseKey, increaseKey, remove
- Binary heap
  - Definition
  - Implementation as array
  - Operations and their running times
  - BuildHeap and its analysis
- Mergeable heaps
  - Binomial heap
    - Definition
    - Binomial tree
    - Algorithms and running times for above heap operations and Merge

Sorting (know algorithm and analysis of all sort methods mentioned below)
- InsertionSort, HeapSort, MergeSort, QuickSort
- Lower bound on comparison sorting
- Linear sorting: CountingSort, BucketSort
- External sorting: External MergeSort

Disjoint sets
- Definition of equivalence relation and equivalence class
- Array-based forest-of-trees representation
- Operations: find, union
- Approaches
  - Union by size, height, rank
  - Path compression
- Analysis: Know worst-case and average case performance of approaches