CS 171
Review for Test 1

Here’s a (partial) list of topics and review problems for the first test. Use this as a reference to begin preparation for the first test.

Chapter 1: Introduction

1. Vocabulary & Definitions
   (a) Connectivity Problem
   (b) QuickFind
   (c) QuickUnion
   (d) WeightedQuickUnion
   (e) Path Compression by Halving
   (f) Full Path Compression

2. Concepts, Skills, & Theory
   (a) Be able to describe the Connectivity Problem and some of its applications.
   (b) Understand the relationship between a solution to the Connectivity problem and the Union-Find Algorithms.
   (c) For each of the solutions presented for the Connectivity Problem (QuickUnion, QuickFind, etc):
      i. Know the data structures used by the algorithm
      ii. Know how the algorithm processes the data
      iii. Big-Oh time of the algorithm
      iv. Worst-, Average-, and Best-case scenarios for each algorithm

Chapter 2: Principles of Algorithm Analysis

1. Vocabulary & Definitions
   (a) Average-case, Worst-case, Best-case performance & bounds.
   (b) CS related functions: $\lfloor x \rfloor, \lceil x \rceil, \lg, N!, \text{Fibonacci & Harmonic numbers}$.
   (c) Asymptotic Upper Bounds, Asymptotic Tight Upper Bounds, $O(f(N)), \theta(f(n))$, “Asymptotic Algebra”
   (d) Recursion & Recurrence Relations
   (e) Sequential Search & Binary Search

2. Concepts, Skills, & Theory
   (a) Understand the main elements of and purpose for Algorithm Analysis
   (b) Understand distinction between average-, worst-, and best-case performance, as well as their relation to guarantees, predictions, and limitations.
   (c) Know the common Big-Oh’s and their order of growth.
   (d) Be able to determine the asymptotic (tight) bounds of combinations of functions.
   (e) Given a for or while loop, be able to answer some Big-Oh questions about their performance (Programming HW 1).
   (f) Given a recursive method or function, be able to derive a recursive relationship that models the method.
   (g) Be able to “solve” a recursion.
(h) Understand the basics of Sequential & Binary Search, explain their best-, average-, and worst-case performance (and Big-Ohs).

Chapter 3: Elementary Data Structures

1. Vocabulary & Definitions
   (a) Data Type, Data Member, Method
   (b) Arrays
   (c) Linked List, Node
   (d) Compound Data Structures

2. Concepts, Skills, & Theory
   (a) Understand the definition of a Data Type and how data types are manifested in Java.
   (b) Understand Arrays and how the various ways they are represented in Java.
   (c) Understand Linked Lists and all its variations (including, but not limited to Table 3.1), and how Linked Lists can be created in Java.
   (d) Be able to construct Compound Data Structures from basic data types.

Chapter 4: Abstract Data Types

1. Vocabulary & Definitions
   (a) Abstract Data Type (ADT)
   (b) Interface
   (c) Client
   (d) Generalized Queues, Stacks, (FIFO) Queues
   (e) Postfix, Infix expressions
   (f) Adapter Class
   (g) First Class ADT
   (h) Cloneable

2. Concepts, Skills, & Theory
   (a) Understand the definitions of an ADT, Interface, and Client, as well as their relationship to one another.
   (b) Understand the motivation/purpose for using ADTs.
   (c) Understand how ADTs can be represented in Java, and the distinction between Java’s Interface and an ADT.
   (d) Given a description, be able to create an ADT.
   (e) Understand when variables in Java store values and when they store references, and how this relates to the Cloneable interface
   (f) Know the various types of Generalized Queues, their applications, and their strengths and weaknesses.