The Final will be Thursday, July 30, from 12:30-2:20 PM in ITT 328 (our normal classroom). It will be closed-book and notes, except for one 8.5” x 11” sheet of paper containing any notes that you want. (Yes, you can use both the front and back of this piece of paper.) About 75% of the test will cover the following topics (and maybe more) since Test 1, and the remaining 25% will be comprehensive (mostly big-oh analysis and general questions about stacks, queues, and heaps).

**Chapter 16: Lists**
Terminology: head, tail, index
Types of Operations and interfaces: Index-based, content-based, position-based
Applications of lists (the general idea only): heap-storage management, organization of files on disk
Indexed-list implementations and analysis: Array-based and singly-linked structure
Positional-list implementations and analysis: Array-based and doubly-linked structure
Iterators and Python implementation
Case Study: Sorted List implementation using an ArrayIndexedList

**Chapter 17: Recursion** (mostly sections: 17.1 and 17.3)
Recursive sorts analysis and implementation: quick sort and merge sort,
Recursive backtracking

**Chapter 18: Trees**
Terminology: node, root, child, parent, siblings, leaf, interior node, edge/branch, descendant, ancestor, path, path length, depth/level, height, subtree
General and binary tree recursive definitions
Tree shapes and their heights: full binary tree, perfectly balanced binary tree, complete binary tree
Applications: parse tree, heaps, binary search trees, expression trees
Traversals: inorder, preorder, postorder, level order
Binary tree ADT: interface, processing, implementation
Binary search tree ADT: interface, implementation
AVL tree ADT: interface, implementation

**Chapter 19: Sets and Dictionaries (Hashing)**
Sets: applications, implementation
List implementation of sets and dictionaries; their analysis
hashing terminology: hash function, hash table, collision, load factor, chaining/closed-address/external chaining, open-address with some rehashing strategy: linear probing, quadratic probing
hashing implementation of dictionaries and sets

**Chapter 20: Graphs**
Terminology: vertex/vertices, edge, path, cycle, directed graph, undirected graph
Graph implementations: adjacency matrix and adjacency list
Graph traversals/searches: Depth-First Search (DFS) and Breadth-First Search (BFS)
General Idea of the following algorithms: topological sort, Dijkstra’s algorithm (single-source, shortest path), Prim’s algorithm (determines the minimum-spanning tree)
You should understand the graph implementations and algorithms listed above. You should be able to trace the algorithms on a given graph.