S-Q 1998 - Hourly 1
July 8, 1998

Name:

Email:

Test Guidelines

• Do not begin the test until you are told to do so.

• This test is closed-book and closed-notes.

• Answer the questions that you find easiest first. There are five questions on the test. The total number of points on the test is 50, and the point value of each question corresponds roughly to the number of minutes you should spend on each question. It is very likely that some of the questions will take you more or less time to answer than our estimates; do not get bogged down on one part of the test.

• Show your work. An incorrect answer that is based on a sound line of reasoning and shows clear understanding (but fails to be correct because of a relatively minor error) may receive more credit than an unsubstantiated statement that happens to be correct.

• Do not feel compelled to fill every square inch of space with writing. There should be plenty of space below each question. If necessary, use the back of the previous page. If this is still not sufficient, ask for scrap paper.
1. (5 points) Draw the binary search tree that would result from inserting the following numeric keys, in the given order, into a binary search tree that is initially empty. Use the ordinary algorithm for insertion (as described in lecture and the reading).

2, 7, 6, 20, 0, 11, 9, 13, 3, 1
2. (10 points) Beginning with an empty queue, draw the *tail queue* (a queue implemented by a circular singly-linked list) data structure that would result from each of the following operations (starting with the empty queue, you should have five separate drawings). Be sure to indicate which direction the pointers in your data structure are pointing.

(a) **ENQUEUE (10)**
(b) **ENQUEUE (4)**
(c) **ENQUEUE (12)**
(d) **DEQUEUE**
3. (15 points) Your co-worker has forgotten how to delete elements from a binary search tree, and has invented a new algorithm to accomplish this task. The new algorithm to delete key \( x \) from a binary search tree \( T \) is:

- Create a new binary search tree named \( N \), initially empty.
- Perform an **in-order** traversal of the binary search tree \( T \). At each node, if the key of the node is **not** \( x \), insert the key and its value into \( N \). Insertion is done in the ordinary manner for binary search trees.
- Destroy \( T \) and replace \( T \) with \( N \).

(a) Assuming that each key in the tree is unique (i.e. no key appears more than once in the tree at any given moment), is this algorithm correct—will the resulting tree contain the correct keys and values, and will the resulting tree be a binary search tree? Defend your claim.
(b) Describe any unusual properties of the trees that are created by this algorithm.

(c) Assume that the destruction of a tree containing \( n \) nodes requires time \( O(n) \), and that tree traversal and binary search tree insertion are done in the manner seen in lecture and the reading. What is the big-O of this delete algorithm? Does it matter whether the key \( x \) is in the tree or not?
(d) Give a better algorithm for doing deletion from a binary search tree— one that will perform at worst at least as well as this algorithm, but will usually perform better. (Hint— the algorithm described in class will suffice, if explained fully.)
4. (12 points) Give an algorithm to rotate a stack containing \( N \) items by \( x \) positions in time \( O(N) \), using only the stack ADT operations Push, Pop, and IsEmpty. You should assume that each of the stack ADT operations requires time \( O(1) \).

The rotation of a stack by \( x \) positions means that the “top” \( x \) values are moved to the “bottom” of the stack.
5. (8 points) Prove by induction that $n^3 + 2n$ is divisible by 3, for all $n \geq 0$. 