CSE 373 Problem Set 2 Due: April 20, 1998

1. The Polynomial Abstract Data Type, together with its linked list implementation, is described on pages 52-54 of the text. In this implementation a polynomial is represented by a linked list with a node for each term. The node has a field for the coefficient of the term, a field for its exponent and a field for a pointer to the next node. The terms are arranged in order of decreasing degree. Figure 3.22 gives an example.

   (a) Describe in pseudocode an algorithm for multiplying two polynomials represented by linked lists. You may assume that List Mergesort or List Quicksort is available as a primitive function.

   (b) Determine the execution time for your algorithm when the first polynomial contains n terms and the second polynomial contains m terms. Assume that the time to sort a list of N elements is $\Theta(N \log N)$. You should aim for an execution time of $\Theta(mn \max(\log n, \log m))$.

   Your first programming assignment will be to write a program for polynomial multiplication based on your pseudocode.

2. Convert the infix expression $f+(a+b(c+d*e))+g$ to a postfix expression.

3. Certain strings consisting of operands and occurrences of the operators $+$ and $*$ are valid postfix expressions. The definition is as follows:

   (a) For any operand a, a is a valid postfix expression;

   (b) If A and B are valid postfix expressions then the string $AB+$ and $AB*$ ($AB+$, for example denotes the string A, followed by the string B, followed by the symbol $+$)are valid postfix expressions;

   (c) No string is a valid postfix expression unless its being so follows from the above two rules.

   Examples of valid postfix expressions are $aa+$, $abc+*$, and $abcd*c+b*++$. Examples of strings that are not valid postfix expressions are $a+$, $bcd*$, and $bcc+*a++$. Give a simple rule for determining whether a string is a valid postfix expression.

4. Exactly how many nodes does a binary tree with n leaves have, where each internal node has exactly two children? Internal nodes are those nodes that are not leaves.

5. Give a recursive algorithm $Count(r, a, b)$ which determines the number of keys in a binary search tree whose values are greater than or equal to a and less than or equal to b, where a and b are integers. Here r is a pointer to the root of the tree. You may assume that no two keys in the tree

are equal. Your algorithm should avoid exploring subtrees that cannot possibly contain values lying between a and b.

6. Suppose we delete a key from a binary search tree and then insert the same key back into the tree. Give an example showing that the tree after the insertion is not necessarily the same as the tree before the deletion.