Data Structures, Fall 2003, midterm exam

(作答在答案卷上題目不用繳回)

1. (15 pts) Answer the following short questions:
    1. What is an *abstract data type*? Give an example.
    2. What is a *data structure*? Give an example.
    3. What do we mean by a binary search tree to be `*balanced*'?
    4. Design a data structure to store the polynomial $5x^{10}-2x^{7}+3x^{3}+100$ efficiently.
    5. Suppose we use the idea of *exclusive-or* to implement doubly-linked list. Explain how it works. That is, what is stored in the pointer field of a node? Give an example.

2. (10 pts) Show the stack and output after each step in the *infix-to-postfix* conversion process. The infix expression is $1*2+(3+4)*5-6$

3. (20 pts) For each of the following questions, select your answer from one of θ(1), θ(log n), θ(n), θ(n log n), θ(n^2). No explanation is needed. Do not guess, 0 for no answer, -1 for wrong answer.

1. The ***minimum*** *time* to delete a key from a binary search tree with *n* nodes.
2. The ***maximum*** *time* to delete a key from a binary search tree with *n* nodes.
3. The ***minimum*** *time* to delete a key from an AVL tree with *n* nodes.
4. The ***maximum*** *time* to delete a key from an AVL tree with *n* nodes.
5. The ***maximum*** *time* to find a key from a splay tree with *n* nodes.
6. The ***minimum*** *time* to find a key in a hash table of size *n* using linear probing.
7. The ***maximum*** *time* to find a key in a hash table of size *n* using quadratic probing.
8. The ***maximum*** *time* to insert a key into an unsorted array of size *n*.
9. The ***maximum*** number of *rotations* needed to perform an *insertion* on an AVL tree of *n* nodes.
10. The ***maximum*** number of *rotations* needed to perform a *deletion* on a splay tree of *n* nodes.

4. (10 pts) Determine the minimum and maximum possible height of an AVL tree with **50** nodes. Explain in sufficient detail how you obtain your answer.

5. (25 pts) Consider the tree T shown in Figure 1. Answer the following questions:
    1. (5 pts) Treat T as a simple binary search tree. Delete 80 from the tree and show the result.
    2. (5 pts) Treat T as a splay tree. Show the result of accessing 54.
    3. (10 pts) Give the preorder, inorder and postorder traversals of T.
    4. (5 pts) By coloring the nodes, is there a way to make T a red-black tree? Why?

6. (5 pts) Show how the following data would be stored in a hash table of size 10 using quadratic probing, assuming that the hash table starts with index 0.

$$h(x) = x \% 10$$

Data: 15, 20, 25, 89, 36, 75, 99, 0, 47, 42

(Note: *x % 10* denotes the remainder of x divided by 10.)

7. (10 pts) Consider the 2-3-4 tree H shown in Figure 2.

1. Insert ***j*** into the tree in a top-down fashion.
2. Draw the red-black tree representation of H.

8. (5 pts) Below is some code to traverse a generalized linked list. It is much like the traversal algorithm we discussed in class, except that we have added an extra print statement that print a B at certain points. **What will it print if we call it for the list L shown below?** The nodes in the list have data, next, and atom fields defined as follows. In the figure, the first field in each node is the data field, and the second field is the next field. The atom field is not shown explicitly, but a node is an atom if and only if it has a digit in its data field.