Data Structure and Algorithm I

Homework #3

Due: 5pm, Thursday, November 4, 2010

Submit the answers for problem 2-4 (paper assignments) through the CEIBA system (electronic copy) or to the TA in R432 (hard copy). You also need to submit the answers of problem 1 (programming assignment) through the CEIBA system.

Problem 1. (40%)

In section 5.8 of the textbook, it was shown how to merge k ordered sequence into a single ordered sequence by using the loser tree. In this problem, please write a program that implements the loser tree. Note that you should consider the case that k is not a power of 2. Your program will need to output the loser tree in each step (each time you move one number from one of the k ordered sequences to the final sequence) using the list representation (see 5.1.2.1, page 194, in the textbook). After the merging operation is completed, output the final merged ordered sequence.

Your program should take the input from the standard input device (stdin) and please use the following input/output format:

Input format:

The first input line is the number of ordered sequences, k, to be merged. k can be stored in a 32-bit signed integer variable.

The following k input lines are the k ordered sequences. Each number in those lines can be stored in a 32-bit signed integer variable.

Example:

```
7
3 5 89
4 9 15 35
3 4 7 8 9
4 9 24
35 66 77 78
34 76 86
```

```
17 27 32 77 95
```

Output format:

If the total number of numbers in the original k sequences is n, then your output should have $n+1$ lines. In the first n lines, please output the list representations of the loser tree in each of the n steps. In the last line, please output the final merged sequence. Example:

```
<List representation of the loser tree in the first step>
<List representation of the loser tree in the second step>
...
<List representation of the loser tree in the n-th step>
3 3 4 4 4 5 7 8 9 9 9 15 17 24 27 32 34 35 35 66 76 77 77 78 86 89 95
```

You must upload your homework in the format of a compressed zip file to the CEIBA, and the zip file should include the following three files:

1. The source code (.c file),
2. A shell script to compile the source (.sh), and
3. A document in PDF format to describe how your program/algorithm works.

Your score of 40% is divided into two parts: correctness of the list representations (9% for each test case) and the merged sequence (2% for each test case) in 3 test cases (33%) and explanations in the document(7%).

Problem 2. (24%) Suppose that we have the following key values: $7, 16, 49, 82, 5, 31, 6, 2, 44$.

1. Assume that we want to create a max heap using the key values shown above. Please draw 3 trees which represent the contents of the max heap after each of the last 3 insertions. (The insertions are performed with the exact order shown above). (6%)
2. Assume that we want to create a min heap using the key values shown above. Please draw 3 trees which represent the contents of the min heap after each of the last 3 insertions. (The insertions are performed with the exact order shown above). (6%)

3. Assume that we want to create a binary search tree using the key values shown above. Please draw 3 trees which represent the contents of the binary tree after each of the last 3 insertions. (The insertions are performed with the exact order shown above). (6%)

4. Please write down the sequence of visits when performing preorder, inorder, and postorder traversals of the max heap after all values are inserted in 1. (6%)

Problem 3. (18%) Heaps are frequently used to implement priority queues. In this problem, we ask you to explore the possibility of using a binary search tree to represent a max priority queue.

1. Please write down your $pop()$ and $push()$ functions. Describe the algorithm for each function with any language (English, Chinese, C, pseudo code, etc.). (12%)

2. Which data structure (heaps or binary search trees) would you use to represent a priority queue? Please explain. (6%)

Problem 4. (18%) A 4-nary tree is defined as a finite set of nodes that consists of a root and either no subtree or 4 disjoint sub-4-nary-trees.

1. Explain how you would implement a data structure to represent a 4-nary tree. (6%)

2. Prove that when the number of leaf nodes in a 4-nary tree is N, the number of all nodes is $\frac{4N-1}{3}$. (6%)

3. What is the maximum and minimum number of nodes when the height of a 4-nary tree is h? Please provide explanations for both answers. (6%)