Submit the answers for problem 2-5 through the CEIBA system (electronic copy) or to the TA in R432 (hard copy). You also need to submit the answers of problem 1 through the CEIBA system. If you choose to submit your writing problems through CEIBA,

1. please combine the answers of all writing problems into only one file in the doc/docx or pdf format, e.g., hw4_writing.pdf; otherwise, you will only get the score of one problem (the one that the TA chooses);

2. please submit your programming report together with your source codes (in the zip archive); if the programming report is submitted with the answers of the writing problems, you will NOT get the score for the report; and

3. please write down your name and school ID in the header of both of the above two documents, e.g., b98902xxx Your_Name

Problem 1. (40%)

In section 10.3.4 of the textbook, it was shown how to insert value into a Red-Black tree. In this problem, you are required to implement the insertion operation of Red-Black Tree. Your program will need to output the preorder traversal of Red-Black tree after each insertion. Your program should take the input from the standard input device (stdin) and please abide by the following input/output format:

Input format:

There are n different positive values that are inserted sequentially and seperated by space. Example:

```
7 3 89 4 9 15 35 8 24
```

Output format:

Your output should have n lines.

The i-th line represent the Red-Black tree in preorder traversal after the i-th input value

insertion.

Notice that you should put a minus("-") before the black node value output.

Example:

```
-7
-7 3
-7 3 89
-7 -3 4 -89
-7 -3 4 -89 9
-7 -3 4 -15 9 89
-7 -3 4 15 -9 -89 35
-7 -3 4 15 -9 8 -89 35
-7 -3 4 15 -9 8 -35 24 89
```

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 (with 4 test cases)(32%) and explanations in the document(8%).

Problem 2. (10%)

1. Refer to Figure 8.7 of the textbook, into an empty dynamic hash tables with directories. The hash table uses buckets that have 2 slots each. Insert elements A0, A1, B0, B1 ,C1, C2, C3 ,C5 (in this order). Show your steps.(5%) - *you can refer Figure 8.8 of the textbook -*

2. Refer to Figure 8.7 of the textbook, into an empty directoryless dynamic hash tables. The hash table uses buckets that have 2 slots each. Insert elements A0, A1, B0, B1 ,C1, C2, C3 ,C5 (in this order). Show your steps. (5%) - *you can refer Figure 8.9 of the textbook -*

Problem 3. (25%)

1. Into an empty min leftist tree, insert elements with priorities 20, 10, 5, 18, 6, 12, 14, 4, and 22(in this order). Show the min leftist tree following each insert. (5%)

2. Delete the min element from the final min leftist tree of part (1). Show the resulting min leftist tree. (5%)

3. Into an empty min B-heap, insert elements with priorities 20, 10, 5, 18, 6, 12, 14, 4, and 22(in this order). Show the resulting B-heap. Show how you arrived at this final B-heap. (5%)

4. Delete the min element from the final B-heap of part (3). Show the resulting B-heap. Show how you arrived at this final B-heap. (5%)

5. Prove that the binomial tree B_k has 2^k nodes, $k \geq 0$. (5%)

Problem 4. (10%) For the Fibonacci numbers F_k and the numbers N_i of Lemma 9.4, prove the following:

1. $F_h = \Sigma_{k=0}^{h-2} F_k + 1$, $h > 1$ (5%)

2. Use (a) to show that $N_i = F_{i+2}$, $i \geq 0$. (5%)

Problem 5. (15%)

1. Start with an empty red-black tree and insert the following keys in the given order: 20, 10, 5 ,30 , 40, 57, 3, 2, 4, 35, 25, 18, 22, 21. Draw figures similar to Figure 10.18 of the textbook depicting your tree immediately after each insertion and following the rebalancing rotation or color change (if any). Label all nodes with their color and identify the rotation type. (10%)

2. What is the largest possible number of internal nodes in a red-black tree with black-height k? What is the smallest possible number? (5%)