CS2123 Data Structures Summer 2017

Assignment 4: Binary Trees

Due 7/17/17 by 11:59pm

1. Searching a Binary Tree (7 points)

Write a C function that accepts a pointer to a binary tree and a pointer to a node of the tree and returns the level of the node in the tree.

You do not have to write a full C program, only the function that returns the level of the provided node pointed in the provided tree pointer. Your answer should be in the text document portion of your answers, not as a separate C source file.

2. Constructing a Binary Search Tree (16 points)

Write a C program to perform the following experiment:

Generate 100 random numbers. As each number is generated, insert it into an initially empty binary search tree. When all 100 numbers have been inserted, print the level of the leaf with the largest level and the level of the leaf with the smallest level. Repeat this process 50 times. Print out a table with a count of how many of the 50 runs resulted in a difference between the maximum and minimum leaf level of 0, 1, 2, 3, and so on.

Implement your binary tree using dynamically allocated nodes, not an array. Remember that each of your 50 repetitions needs to print the largest and smallest leaf levels. At the end of all 50 runs, your program will print a summary of level differences and the # of runs. Here an example of the output format:

Level Difference	# Runs
0	3
1	5
2	6
3	1
4	0
5	3
⋮	⋮

Use the following function to generate a random int from 1 to 1000 for each node you insert into your binary search tree:

```
int generateRandom() {
   return rand() % 1000 + 1;
}
```

Continued on the back ↪

You should initialize the random number generator with the following line in your main function:

```
srand(time(NULL));
```

Note: to use the time() function, you need to include the header file <time.h> at the top of your C source file.

3. Huffman Encoding (7 points)

Encode the following message by hand using Huffman encoding:

"cannercancanacanra"

In addition, show the tree and priority queue after each time a pair of nodes are merged. To make drawing the trees easier you can handwrite your answer this problem. If you do this please scan and submit your answer as a pdf file (there are free to use scanners in JPL).

Deliverables:

Problem 2 should be submitted as one or more C source code files.

Your answer to homework problem 1 should be typed and submitted in a separate document. Your answer to homework problem 3 can be typed or handwritten and should submitted in a separate document.

Archive and submit the files in UTSA's Blackboard system (learn.utsa.edu) under Assignment 4.

Remember:

The program you submit should be the work of only you. Cheating will be reported to judicial affairs. Both the copier and copiee will be held responsible.