

COP 3502 Section 2 Final Exam - Part B (Bin Tree, Hash Table/Heaps, AVL Trees/Tries)

Date: 7/29/2020

Start Time: 4:30 pm EST

End Time: 5:00 pm EST

Directions: Please type up answers in either a Word Document (.doc, docx) or a Text Document (.txt) or scan your written work to a .pdf file and upload your document AND SUBMIT IT to the appropriate assignment in Webcourses COP 3502 Section 2. Don't recopy the questions due to the time constraints, but clearly label which question number your work corresponds to. On the document you submit, put your first and last name in the top left hand corner. On the following line, write "My Exam 1 Part B Answers", centered. Following that, place your answers, numbered, in order (1, 2, 3).

1) (10 pts) Write a *recursive* function that counts and returns the number of nodes in a binary tree with the root `root`, that store an even value. Please use the struct shown and function prototype shown below. (For example, if the tree rooted at `root` stored 2, 3, 4, 8, 13, 18 and 20, the function should return 5, since there are five even values [2,4,8,18,20] stored in the tree.

```
typedef struct node {
    int data;
    struct node* left;
    struct node* right;
} node;

int numEvenNodes(node* root) {
    // Fill in code
}
```

2) (5 pts) Consider inserting the following values into a min heap, in this order: 12, 3, 19, 2, 1. Show the final locations for each value in the array storing the heap. (Recall that we store heaps in arrays using 1-based indexing and typically leave the 0 index blank.) Note: Only the answer will be graded for this question.

index	1	2	3	4	5
value					

3) (10 pts) Write a function that takes in a root node of a trie and returns the length of the longest word stored in that trie. Use the struct given and function prototype given below.

```
typedef struct trienode {
    int isWord;
    struct trienode* next[26];
} trienode;

int maxWordLength(trienode* root) {
    // Fill in code
}
```