Computer Science I (COP 3502) Exam 2 Practice Questions
Topics: Sorting, Binary Trees, AVL Trees, Heaps, Tries and Hash Tables

1) Show the contents of the following array after each iteration of Bubble Sort:

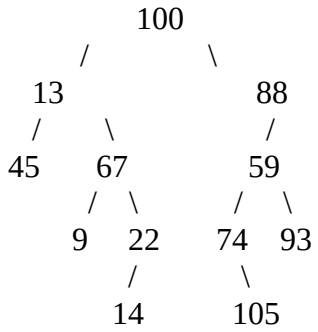
Initial Values	8	3	6	1	7	5	2
1st iteration							
2nd iteration							
3rd iteration							
4th iteration							
5th iteration							
Last iteration	1	2	3	5	6	7	8

2) Show the result of partitioning the array below, using the leftmost element as the partition element. Please use the in-place partitioning algorithm shown in class.

Initial Values	6	9	3	12	5	2	13	18	4	1	7
After Partition											

3) Why does Quick Sort run faster than Merge Sort, on average, in practice?

4) Provide the Preorder, Inorder and Postorder traversals of the following binary tree:

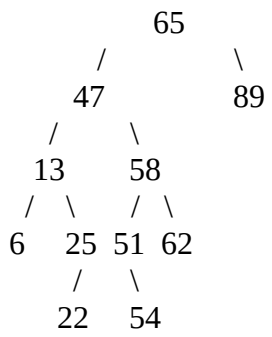


Pre-Order: ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___

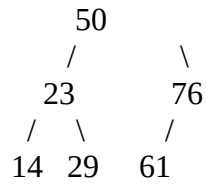
In-Order: ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___

Post-Order: ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___

5) Show the result of deleting 47 from the binary search tree shown below. (Note: There are two possible right answers.)



6) In a binary search tree, consider adding all the values at an even depth from the root, and subtracting all the values at an odd depth from the root. For example, for the tree shown below:

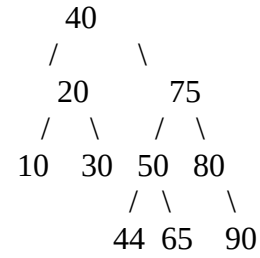


The desired sum would be 50 - 23 - 76 + 14 + 29 + 61 = 10.

Write a recursive function that calculates this adjusted sum for a binary search tree, given a pointer to its root. **(Hint: If I am a tree rooted at 50 above, then the contribution to my sum is the NEGATIVE of the corresponding sum for each of my subtrees.)** Please use the struct and function prototype given below:

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

int getAdjustedSum(bintreenode* root) {









}
```

7) Show the result of inserting the value 60 into the AVL tree below. Put a box around your final answer.



8) Complete the function below so that it counts the number of words in the trie pointed to by root and returns this value. (Note: isWord is guaranteed to be either 0 or 1 for all nodes.) Please use the struct definition and function prototype shown below:

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

int numWords(trie* root) {









}
```

9) Consider implementing a hash table that stores integers, using the linear probing strategy. Assume that the hash table uses the hash function, f, defined below and that size is 23. Show the contents of the table after the following insertions have been made, in the order given: 4362, 999235, 7283624, 8123456, 77, 11111111, 52, 123, 7999999 and 12345675.

```
int f(int n, int size) {
    int res = 0;
    while (n > 0) {
        res = (res + (n%10))%size;
        n = n/10;
    }
    return res;
}
```

index	0	1	2	3	4	5	6	7	8	9	10	11
value												

index	12	13	14	15	16	17	18	19	20	21	22
value											

10) Consider inserting the following items into a minimum heap in the following order: 18, 8, 16, 4, 3, 12, 17, 2, 22, and 7. Show the state of the heap (drawn as a complete binary tree) after the completion of each insertion. Draw a box around each of your answers.