Csci 242. Algorithms and Data Structures **Fall, 2018**

Date: October 9th, 2018

Exam 1: 100 + 20 (optional) points

Name: ______________________

1. Your answer should be fully explained; any sloppy answer without explanation would not get a full point.
2. Your writing should be clear and readable.
3. Any algorithm has to be written in a pseudo code. If an algorithm needed any sub-algorithm to complete it, you also have to write its sub-algorithm(s) in the pseudo code, too.
4. For any array, its index begins from 1: A[1 .. n]

Mark the followings;

Difficulty:

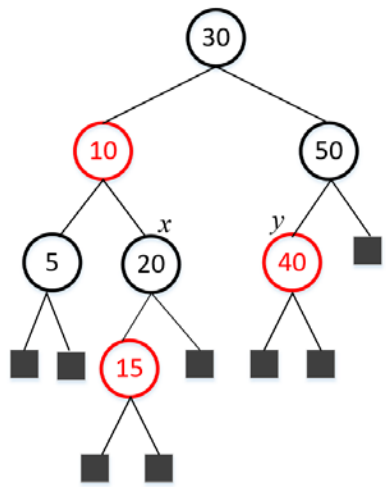
Very Easy: _____ Easy: _____ Moderate: _____ Difficult: _____ Very Difficulty: _____

Time:

Short: ________ Enough: ________ Too Much: _________

Q1. [20] Data Structure

In the given tree,



(1). [5] Give (A) a **height** of a node *x* and (B) a **black-depth** of the node *y*.

(2). [5] Suppose that the colors of the nodes of keys 10, 15 and 40 are also black, i.e. every node has the same color. Is the tree an AVL tree? Justify your answer.

(3). [5] Traverse the tree by ***Preorder Traversal*** and print the keys.

(4). [5] Is the tree a ***binary maximum heap***? Justify your answer.

Q2. [20] Short Answer. Explain your answer clearly.

(1). [5] What is a ***running time* of *deletion*** and that of ***insertion*** operation of ***max-priority queue*** of n elements implemented by a ***heap,*** respectively? Give them in big-Oh(O) notation.

(2). [5] What is the ***maximum* number of *internal nodes*** of a binary ***heap*** of height h, that stores n elements? Explain your answer by drawing such a heap.

(3). [5] Give a **definition** of a ***Red-Black tree*** and describe its **property.**

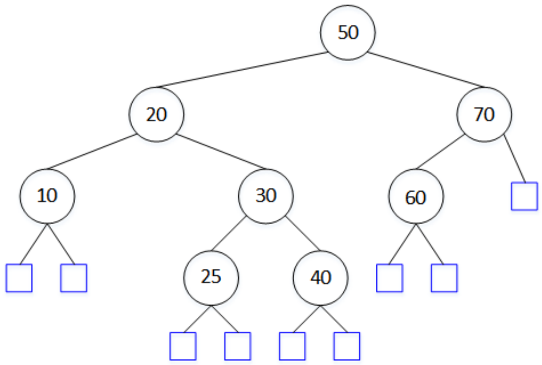
(4). [5] Explain (A) an ***open addressing*** is in the hashing and (B) give *three methods* for open addressing.

Q3. [10] Write a ***recursive*** algorithm, **Tree-Sum(*v*),** that returns the sum of the ***keys of internal nodes*** in the tree rooted at a node *v*. A tree is implemented as a *linked list structure* and all of the external nodes are dummy with no key.

Q4. [10] (A) Write an insertion algorithm **put(*k*)** which **inserts** a key *k* to the hash table A of size N using the hash function *h* and handling a collision with **linear probing**. (B) Then, insert a key 17 to the given hash table of size 11

0	1	2	3	4	5	6	7	8	9	10
	23			15		28	7	39	18	

Q5. [10] Draw the ***AVL tree*** after **deleting** the key **50** from the following AVL tree.
Show each step of deletion & rotation(s) to restore the AVL property for the final tree.



Q6. [10] Write a ***recursive algorithm*** that computes $\sum_{i=0}^{n} x^i$ in $\mathbf{O}(n)$ time.

Q7. [20] Min-Heap

In the given array A[1.. n] = [15, 9, 7, 12, 5, 6, 3],

(1) [10] A heap can be constructed by merging two heaps. Write an in-place **recursive algorithm BottomUpHeap(A)** that constructs a **min-heap** in the array A[1 .. n] by merging two sub-heaps with a new key. Suppose that the number of stored keys $n=2^h-1$ where h is the height of a heap.

(2) [10] Construct a min-heap in the given array A by BottomUpHeap. Show the **final min-heap** in the **array A**.

(3) [10, optional] What is the ***running time*** of **BottomUpHeap** in its asymptotic ***upper bound,*** big-Oh (O)? Write its recurrence equation of T(n) and solve it.

Q8. [10, optional] Analysis of Algorithm

For an algorithm whose total running time is $T(n) = n^2 + 2n - 8$, (A) give its ***asymptotic upper bound*** in ***big-Oh*** notation (**O**) and (B) prove it by the definition of big-Oh.