CST 370 Design and Analysis of Algorithms
Midterm – II (SP'20)

Name: ____

Four-digits ID: _____________________________________

"On my honor, I have neither given nor received unauthorized aid in doing this assignment."

Signature (Write Your Name)_______________________________

- Do not start until told to do so.
- Look over all the questions and observe their point values before you start.
- Use your time wisely—make sure to answer the questions you know first.
- **Read the questions carefully.**

1. (2 points) Consider the following master theorem:

$T(n) = aT(n/b) + f(n)$ where $f(n) \in \Theta(n^d)$, $d \geq 0$

Master Theorem: If $a < b^d$, $T(n) \in \Theta(n^d)$
If $a = b^d$, $T(n) \in \Theta(n^d \log n)$
If $a > b^d$, $T(n) \in \Theta(n^{\log_b a})$

Based on the theorem, select the correct time efficiency for each T(n). You have to **select and write your answer among 1, 2, 3, 4, and 5 clearly**.

(a) T(n) = 2 * T(n/4) + 4n + 7

1. $\Theta(n^2)$
2. $\Theta(n*log\ n)$
3. $\Theta(n)$
4. $\Theta(n^{\log_4 2})$
5. None of the above.

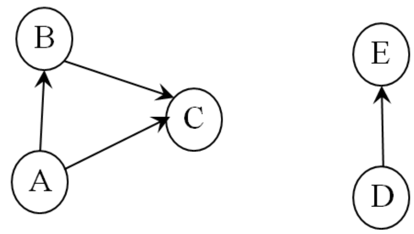
Your answer: ___________

(b) T(n) = 4 * T(n/2) + $3n^2$ + 5n

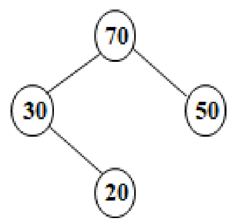
1. $\Theta(n^2)$
2. $\Theta(n*log\ n)$
3. $\Theta(n)$
4. $\Theta(n^{\log_4 2})$
5. None of the above.

Your answer: ___________

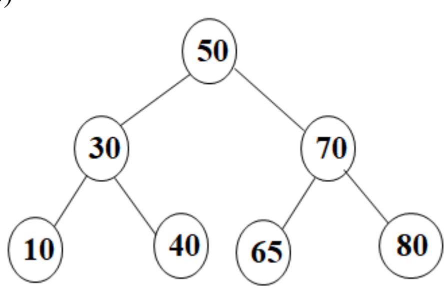
2. (3 points) (a) Is the following graph a DAG (= directed acyclic graph)? (Yes/No)



(b) Is this an AVL tree? (Yes/ No)



(c) Is this a 2-3 tree? (Yes/ No)



3. (1 point) The following algorithm is designed to calculate the number of leaves in a binary search tree. Is this algorithm correct? (Yes / No)

```
Algorithm LeafCounter(T)
//Input: A binary search tree T
//Output: The number of leaves in T
if (T == NULL)
    return 0
else
    return LeafCounter(T_LEFT)+ LeafCounter(T_RIGHT)
```

4. (5 points) Consider the following algorithm.

```
// Assume that n is a positive integer (= i.e. n >= 1), and A[1..n] is a global array.
// Note that the index of array A starts from one, not zero.
// And also, don't forget the array A in the algorithm is global.

Algorithm DoSomething (n)
if (n = 1)
    Print the current content of the whole array A in a single line;
    Move the cursor to the next line.
else
    for i ← 1 to n do
        DoSomething(n – 1);    // Recursive call.
        if n is odd
            swap A[1] and A[n];
        else
            swap A[i] and A[n];
return;
```

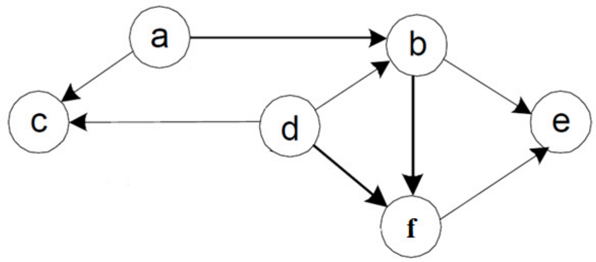
(a) Present execution result of the algorithm where an array **A** has "**5**"and *n* is 1.

(b) Present execution result of the algorithm where an array **A** has "**7, 3**"and *n* is 2. Note that **the sequence of output results is important**. Thus, you have to describe your answer clearly.

5. (2 points) Consider the quicksort algorithm covered in the class. Present the **result of first partitioning operation** for the list "**50 40 30 20 10 70**". In other words, you have to conduct the operation until the indexes i and j meet and cross over. After that, the pivot value should be swapped. For the problem, you should use the first number, **50, as a pivot** for the partitioning. For the problem, **do not present the intermediate steps**. **Just write the sequence of numbers** after the fist partitioning operation.

Your answer: ____

6. (3 points) For the following graph, you are going to conduct the **topological sorting** using the **source removal algorithm** (= Kahn's algorithm).



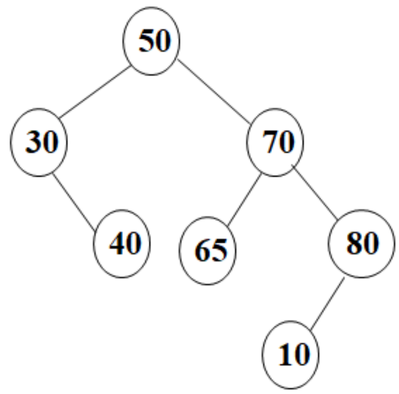
(a) Present the initial **"in-degree" values** of all vertices as we discussed in the class.

a	
b	
c	
d	
e	
f	

(b) Present the topological order of the graph using the source removal algorithm as you learned in the class. For the problem, you have to follow our convention of alphabetical order.

Topological Order: ____

[Note] Before solving the problem 7, 8, 9, and 10, read the following description carefully.
In the problem 7, 8, 9 and 10, you have to present the result trees in the **level-by-level order**. This is an example of level-by-level order for a sample tree below. Note that the root value 50 is the level 0. Then, its children (= 30 and 70) should be the level 1. Also, because there's no value in the level 4 and 5, we use "NONE" to indicate them.



A Sample Tree

Level 0	**50**
Level 1	**30, 70**
Level 2	**40, 65, 80**
Level 3	**10**
Level 4	**NONE**
Level 5	**NONE**

Level-By-Level Order

7. (5 points) (a) Consider a binary tree with three nodes with the values 10, 20 and 30 in such a way that the inorder and preorder traversals of the tree yield the following lists:

10, 30, 20 (inorder)
30, 10, 20 (preorder).

Note that the problem is asking to **consider only one binary tree**. For the problem, **do not draw the result in the word file**. Instead, **write the values of the result tree level-by-level order**.
If you think that it's not possible to have a binary tree with the given information, explain why.

Level 0	
Level 1	

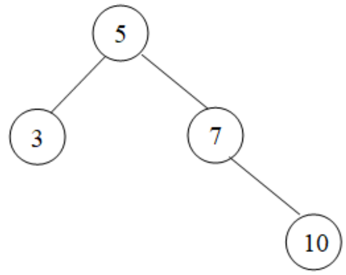
(b) Consider a binary tree with six nodes with the values 10, 20, …, 60 in such a way that the inorder and postorder traversals of the tree yield the following lists:

30, 60, 50, 20, 10, 40 (inorder)
30, 60, 20, 40, 10, 50 (postorder)

Note that the problem is asking to **consider only one binary tree**. If you think that it's not possible to have a binary tree with the given information, explain why.

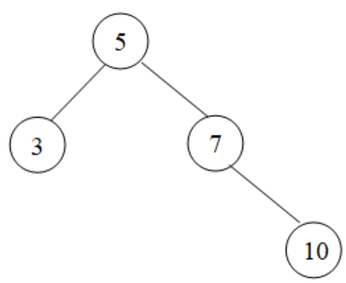
Level 0	
Level 1	
Level 2	
Level 3	

8. (4 points) (a) Assume that you have an AVL tree like below. Add a node with the value **20**. After that, present the resulting AVL tree using the level-by-level order. For the problem, **do not draw the result tree in the word file**. Instead, **write the values in the result tree level-by-level order**.



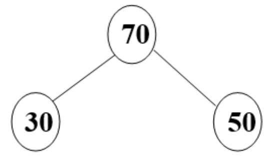
Level 0	
Level 1	
Level 2	
Level 3	

(b) Assume that you have an AVL tree like below. Add a node with the value **8**. After that, **write the values in the result tree level-by-level order**.



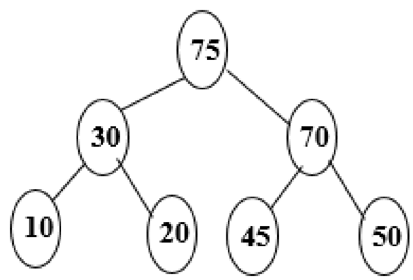
Level 0	
Level 1	
Level 2	
Level 3	

9. (2 points) (a) Add 55 to the following **max heap**. After that, **write the result max-heap using the level-by-level order**.



Level 0	
Level 1	
Level 2	

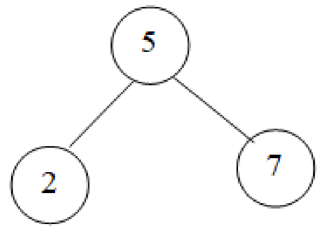
(b) Delete the max value from the following max heap. After that, **write the result max-heap using the level-by-level order**.



Level 0	
Level 1	
Level 2	
Level 3	

10. (3 points) Consider a **2-3 tree** as below. Add the following four numbers to the tree one by one. After that, **write the result 2-3 tree using the level-by-level order**.

10, 11, 8, 9



Level 0	
Level 1	
Level 2	
Level 3	