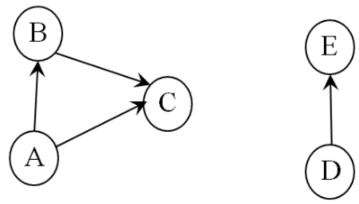
CST 370 Design and Analysis of Algorithms
Summer A 2020
Final Exam

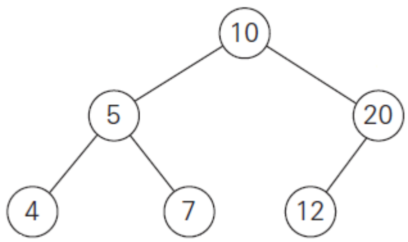
Name: Carlos Orduna
Four-digits ID: 0112

- Test time is **2 hours and 30 minutes**.
- Note that there are **13 problems in the final exam**.
- This is a **closed book exam**. You **can't use a calculator** during the exam. However, as a reference during the exam, you can prepare "two pages (= total of 4 sides)" **cheat sheet**. The cheat sheet can be typed or hand-written.
- **Blank paper can be used for your calculation** when solving problems.
- If possible, enter your answers directly into the Word file to increase the readability of your answers. However, if it is too difficult or time consuming to type in a Word file, write down your answer on paper. Then, take a picture and insert the picture into the Word file.
- During the exam, **turn on the video**, but **turn off the audio**. We will record the whole zoom session.
- **If you have a question** during the exam, **use "Chat"** in Zoom. I will answer to your question using "Chat".
- When you finish the exam, submit your PDF file (and optional Word file) on the iLearn. But keep the file(s) well in case we need it.
- Use your time wisely—make sure to answer the questions you know first.
- Read the questions carefully.

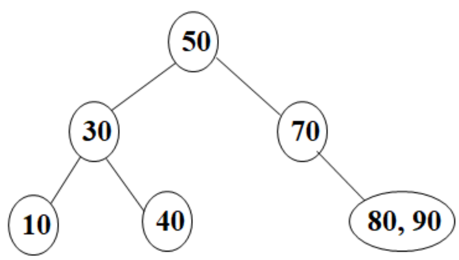
1. (2 points) Is the following graph a DAG? (Yes/No)



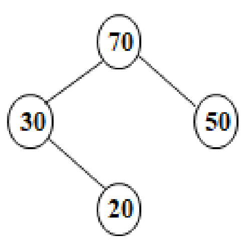
(b) Is this an AVL tree? (Yes/ No)



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



(d) This is a max heap. (True/False)



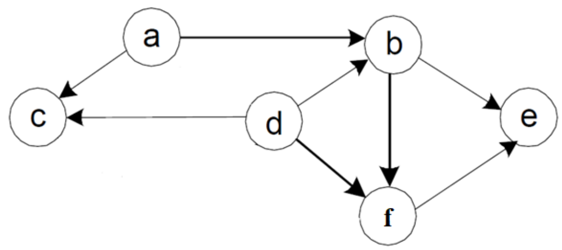
2. (1 point) The following algorithm is designed to calculate the number of leaves in a binary search tree. Is this algorithm correct? (Yes / No)

```
Algorithm LeafCounter(T)
//Input: A binary search tree T
//Output: The number of leaves in T
//        TLEFT is a left subtree of T.
//        TRIGHT is a right subtree of T.
if (T == NULL)
    return 0
else
    return LeafCounter(TLEFT)+ LeafCounter(TRIGHT)
```

3. (3 points) Assume that you have five different data structures like below. In each data structure, there are *n* integer numbers and want to check if a specific number exists or not. In other words, you want to search a number in the data structure.
Write the **worst case time complexity** of the search operation in each data structure using the **Big oh (O) notation**.

Unsorted array	**O(n)**
Sorted array	**O(log n)**
Binary search tree	**O(n)**
AVL tree	**O(log n)**
Hashing (Separate Chaining)	**O(n)**

4. (2 points) Apply the Kahn's algorithm to solve the topological sorting problem for the following digraph. Present the "in-degree" array and the topological order clearly.

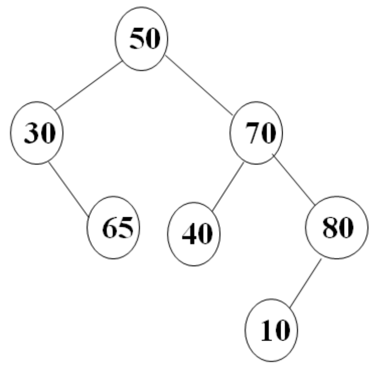


"in-degree" array

Vertex	In-degree
a	0
b	2
c	2
d	0
e	2
f	2

Topological Order: a -> d -> b -> c ->f -> e

[Note] Before solving the problem 5, 6, 7, and 8, read the following description carefully.
In the problem 5, 6, 7 and 8, 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) are in the level 1. Note that because there's no value in the level 4 and 5, we use "NONE" to indicate it.



A Sample Tree

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

Level-By-Level Order

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

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

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** in **level-by-level order**. If you think that it's not possible to have a binary tree with the given sequences, indicate it clearly.

Level 0	**30**
Level 1	**10, 20**

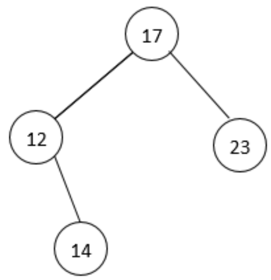
(b) Consider a binary tree with six nodes with the values 10, 20, …, 60 in such a way that the inorder and preorder traversals of the tree yield the following sequences:

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

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 sequences, indicate it clearly.

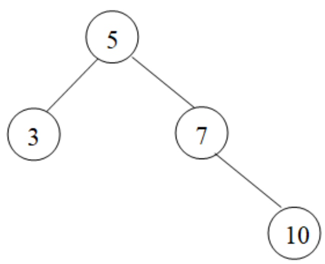
Level 0	**40**
Level 1	**20, 50**
Level 2	**10, 60, 30**
Level 3	**NONE**

6. (2 points) (a) Assume that you have an AVL tree like below. **Add a node** with the **value 15**. After that, present the result AVL tree using the level-by-level order. For the problem, **do not draw the result in the word file**. Instead, **write the values of the result tree in the level-by-level order**.



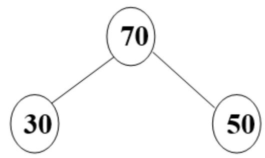
Level 0	**17**
Level 1	**14, 23**
Level 2	**12, 15**
Level 3	**NONE**

(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**.



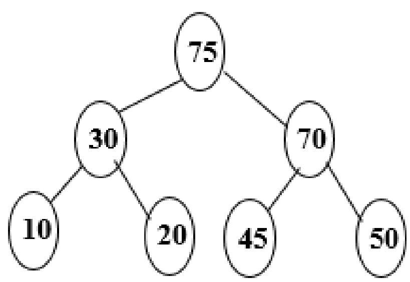
Level 0	**5**
Level 1	**3, 8**
Level 2	**7, 10**
Level 3	**NONE**

7. (2 points) (a) Add 75 to the following **max heap**. After that, **write the result max-heap using the level-by-level order**.



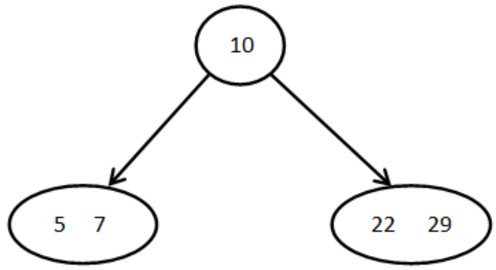
Level 0	**75**
Level 1	**70, 50**
Level 2	**30**

(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	**70**
Level 1	**30, 50**
Level 2	**10, 20, 45**
Level 3	**NONE**

8. (2 points) Assume that you have a 2-3 tree like below. Add two numbers **9** and **27** to the tree one by one. After that, **present the result 2-3 tree using the level-by-level order**



Level 0	**10**
Level 1	**7, 27**
Level 2	**5, 9 , 20, 29**
Level 3	**NONE**

9. (2 points) Assume that you construct a hash table using the **separate chaining** for the following 7 keys:

30, 77, 24, 14, 9, 12, 27

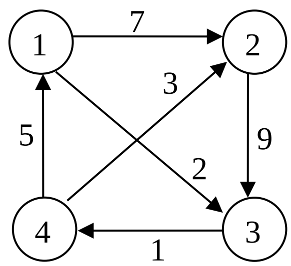
Assume that the hash function is ***h(K) = K mod 5***. For the problem, **do not consider the rehashing**.

(a) After constructing the hash table with the keys, **present all indexes** in the table which **have any collision(s)**. If there's no index with collision, write it clearly.

The indexes with a collision are indexes **2 & 4**.

(b) For the hash table constructed at the above question, assume that you insert a new key value **18**. Is there any collision for the key "**18**"? (Yes / No).

10. (2 points) Apply Floyd's algorithm to find all-pairs-shortest-paths of the digraph defined by the following graph. For your understanding, $D^{(0)}$ is already provided. Fill out $\mathbf{D^{(1)}}$ and $\mathbf{D^{(2)}}$. Note that you **don't need to present and $\mathbf{D^{(3)}}$ and $\mathbf{D^{(4)}}$** for the problem. Also, you can just **use "INF" instead of the infinity symbol** to indicate an infinity value.



$D^{(0)}$ =

	1	2	3	4
1	0	7	2	∞
2	∞	0	9	∞
3	∞	∞	0	1
4	5	3	∞	0

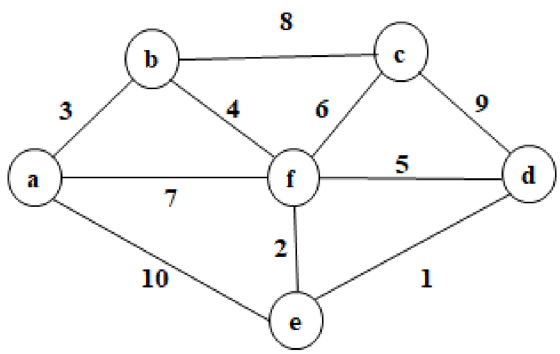
$D^{(1)}$ =

	1	2	3	4
1	0	7	2	Inf
2	Inf	0	9	Inf
3	Inf	inf	0	1
4	5	3	7	0

$D^{(2)}$ =

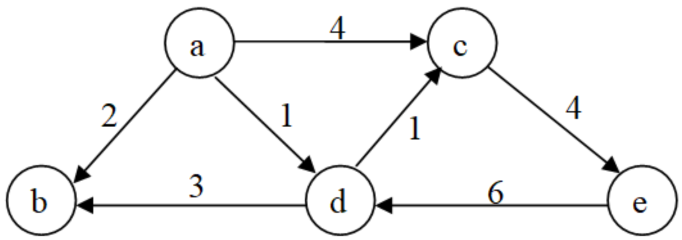
	1	2	3	4
1	0	7	2	Inf
2	Inf	0	9	Inf
3	Inf	inf	0	1
4	5	3	7	0

11. (3 points) Determine the MST using **Prim's algorithm** for the following graph as you learned in the class. For the problem, start the algorithm from the vertex **a,** and it's good enough to **present first three steps (= three rows of the table below)**.



Tree Vertices	Remaining Vertices
a(-,-)	**b(a, 3), c(-, inf), d(-, inf), e(a, 10), f(a, 7)**
b(a,3)	**c(b, 8), d(-,inf), e(a, 10), f(b, 4)**
f(b,4)	**c(f, 6), d(f , 5), e(f, 2)**

12. (3 points) Assume that you are going to solve the single-source shortest-paths problem using the **Dijkstra's algorithm** for the following directed graph. For the problem, you should start from the vertex **a**. Fill out the table as we covered in the class.



V	a	b	c	d	e
a	0_a	2a	4a	1a	Infinity
		2a	1d	1a	Infinity
		2a	1d		4c
		2a			4c
					4c

13. [**Puzzle**] (2 points) Assume that there are **two missionaries** and **two cannibals** in a river. They have to cross the river using a boat which can accommodate up to two people.

One constraint in the problem is that missionaries cannot be outnumbered by cannibals. For example, the situation of one missionary and two cannibals is unacceptable. But one missionary and one cannibal are fine.

Present the **minimum number of crossings of the boat** so that all four can cross the river.

For the problem, you **don't need to explain your idea**. **Just write the minimum number of crossings**.

If they can't cross the river by the boat, write it clearly.

5