United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

FINAL ASSIGNMENT Total Marks: **40** **Spring - 2020**

Course Code: **CSI 217** Course Title: **Data Structures**

Answer all 4 questions.

1. Answer the following questions on **BINARY TREE**.

a. Construct a Binary Search Tree using the values below: [3]
8, 16, 11, 3, 6, 5, 18, 1, 19
(Remember you have to insert these values in the given order)

b. When does the worst case time complexity scenario occur in a BST? Explain with example. [2]

c. Is the tree that you constructed in **1.a.** [2]
i. a Complete Tree?
ii. a Balanced Tree?
You have to clearly explain your answer.

d. Perform the following operations on the tree you constructed in **1.a.** Each operation should be applied on the resultant tree. You must clearly show your steps and draw the state of the BST after each operation. [3]
i. Insert(7)
ii. Delete(18)
iii. Delete(8)

e. Write the pseudocode to perform **Breadth First Search(BFS)** on a Binary Search Tree. [4]

How would you modify your pseudocode in 1.d. to perform **Depth FIrst Search(DFS)** on a Binary Search Tree? Explain your answer.

---

2. Answer the following questions on **HEAP**.

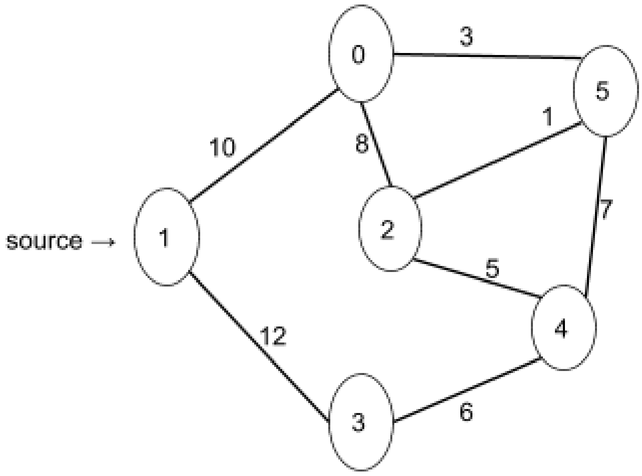
a. Does the array below represent a MIN HEAP? If not then modify it to create a MIN HEAP. [3]

1	11	5	9	4	16	3	6	8

b. Perform the following operation on the heap you have created in 2.a. Each operation should be applied on the resultant tree. [3]
i. Insert(20)
ii. Delete root
iii. Delete root

c. Explain how you would implement a **MIN PRIORITY QUEUE** using heap. [2]

3. Answer the following questions on the **UNDIRECTED WEIGHTED GRAPH** below:



a. Create the **Adjacency Matrix** for the graph above. [2]

b. Write a pseudocode to perform **Breadth First Search** on the graph above. You also have to **include an array** in your pseudocode to **keep track of the parents** of all the vertices during traversal. [4]

c. Show the manual tracing to create the **d[] array** for the graph above using **BFS**. The d[] array is an array that keeps track of distance of each vertex from the source. Show all the calculations and all the extra data structures or arrays that you have used to perform the manual tracing. [4]

4. Answer the following questions on **SET OPERATIONS**.

0	1	2	3	4	5	6	7	8	9
0	1	1	8	3	0	5	1	8	8

a. Perform the following operation on the array above using **QUICK FIND**: [3]
   i. Union(6,1)
   ii. Find(4)
   iii. Connected(1,3)?

b. Perform the following operation on the array above using **QUICK UNION**: [3]
   i. Union(6,1)
   ii. Find(4)
   iii. Connected(1,3)?

c. Suppose the array above was created using the QUICK FIND technique, how would to count the total connected components? [2]
Will you use the same process to count the total connected components if the array was created using Quick Union? Why?

GOOD LUCK!