CS210 Data Structures
(201) Final Exam

Name:____ ID__________________________

Check your section:

☐ Dr. Sara Shaheen at Morning 10 AM ☐ Dr. Sara Shaheen at Afternoon 2 PM ☐ Dr. Sawsan Alhalawani on Sunday ☐ Dr. Sawsan Alhalawani on Monday	☐ Dr. Basit Qureshi ☐ Dr. Muhammad Akour ☐ Dr. Zahid Khan

Instructions:

- This exam contains four questions with multiple parts, on 10+1 sheets of papers
- DONOT detach the scratch-sheet
- Time allowed: 120 minutes
- Closed Book, Closed Notes.
- Use of Calculators is ALLOWED. Use of other computing devices / smartphones etc is strictly prohibited.
- Answer the problems on the exam sheets only. No additional attachments would be accepted.
- **DO NOT write on the backside of a page/sheet**; the back of a page will NOT be graded.
- When the "time is over" is called, it is the students' responsibility to submit his exam to the invigilator. Submitting completed exam 3 minutes after the "time is over" will incur a penalty of **5 points**.

Few gentle reminders:

- If you get stuck on some problem for a long time, move on to the next one.
- You should be better off by first reading all questions and answering them in the order of what you think is the easiest to the hardest problem.
- Keep the points distribution in mind when deciding how much time to spend on each problem.

Question No.	Part a	Part b	Part c	Part d	Student's Score
Question 1 (CLO 1)					/13
Question 2 (CLO 2)					/8
Question 3 (CLO 3)					/16
Question 4 (CLO 4)					/3
Total					/40

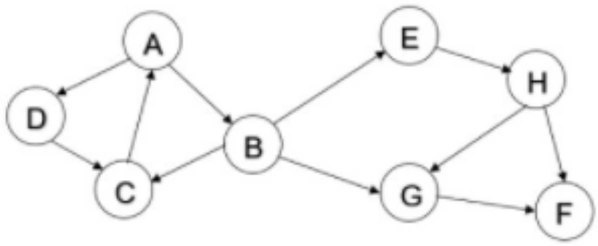
DEPARTMENT OF COMPUTER SCIENCE

CCIS كلية علوم الحاسب والمعلومات COLLEGE OF COMPUTER & INFORMATION SCIENCES

جامعة الأمير سلطان PRINCE SULTAN UNIVERSITY

Question 1. [2 + 4 + 3 + 4= 13 points]

Part a. [2 points] Given the following directed graph, draw/show the <u>adjacency-list</u> to store this graph.



Part b. [4 points] Show the result of the DFS and BFS on this graph (part a). For path resolution, assume a lower order letter in alphabet chronology comes before a higher order (e.g. A comes before Z). Use appropriate Data Structures to support your response. Show your work!

-Depth-First-Search

-Breadth-First-Search

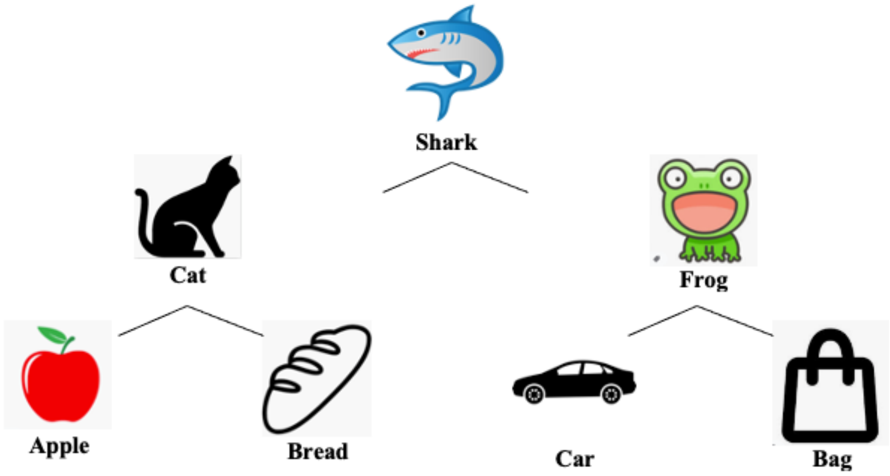
Part c. [3 points] Draw the binary tree given in this array.

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
element	**9**	**3**	**10**	**1**	**6**		**14**			**4**	**7**			**13**

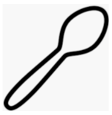
What do you have to do to convert this tree to AVL? Show the tree after appropriate rotations. Identify what type of rotation?

Now insert a new node (8) in this tree. Show the tree after appropriate rotations. Identify what type of rotation?

Part d. [4 points] We need to store images in an efficient data structure. The following is a max-heap of images. Each image is annotated with a String text-label which is to be used to compare items. (example: "BAG" comes before "BAT").



Remove an image from this heap. What image is returned? Show the resulting heap.

Add  **(Spoon) to this heap. Show the resulting heap.**

Question 2. [4 + 4= 8 points]

Part a. [4 points] Answer the following Multiple Choice Questions

Problem	Answer
1. Given the following sequence: {2, 3, 5, 6, 9, 11, 15}. Which sorting algorithm will take the most time (number of comparisons)? Consider typical implementations of sorting algorithms. **a.** Insertion Sort **b.** Selection Sort **c.** Heap Sort **d.** Merge Sort	
2. Given the following sequence: {2, 3, 5, 6, 9, 11, 15}. Which sorting algorithm will run in O(n) time (n comparisons)? **a.** Insertion Sort **b.** Selection Sort **c.** Heap Sort **d.** Merge Sort	
3. Given the following sequence: {2, 3, 5, 6, 9, 11, 15}, looking for 15, the Binary Search algorithm will run in what time? **a.** O(1) **b.** O(log n) **c.** O(n) **d.** O(n log n)	
4. What sorting algorithm takes more space than the others? **a.** Insertion Sort **b.** Selection Sort **c.** Heap Sort **d.** Merge Sort	
5. Which of the following runs fastest? **a.** Searching for an element that is not in a singly linked list **b.** Sorting a pre-sorted array (same order) using heapsort **c.** Searching the value in the root of a AVL tree **d.** Emptying all elements from the stack	
6. Which of the following runs slowest? **a.** Searching for an element that is not in a singly linked list **b.** Sorting a pre-sorted array (same order) using heapsort **c.** Searching the value in the root of a AVL tree **d.** Emptying all elements from the stack	
7. Which of the following runs fastest? **a.** Searching a node in an AVL tree **b.** Searching a node in a skewed Binary Search Tree (all items are identical) **c.** Running Pre-order traversal algorithm on a BST **d.** Searching for an element in a circular linked list	
8. Which Graph implementation is most inefficient in terms of space **$O(V^2)$** **a.** Adjacency List **b.** Adjacency Matrix **c.** Edge-List **d.** None of these	

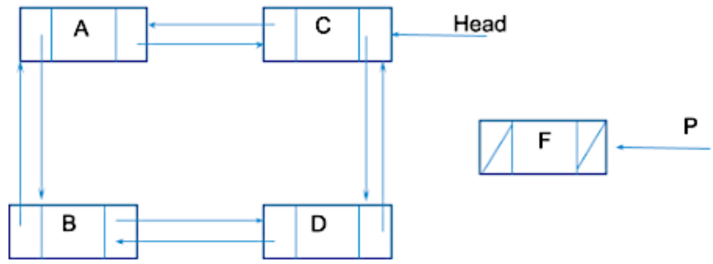
Part b. [4 points] Sort the elements of the following array using top-down merge sort approach. Show all operations (Lo-mid-hi indexes and merge operations).

Cat	Bit	Hat	Mat	Rat	Bat	Hot	Bot	Sit

Note: In case you forgot your ABCs! use this to help you determine the order

Question 3. [4 + 4 + 4 + 4 = 16 points]

Part a. [4 points] The diagram below is a Circular-Doubly Linked List data structure. Provide the necessary code to add the new node pointed by P to the list. It does not matter where you add it. However, the list must be kept circular and doubly linked list.



Part b. [4 points] Assume an implementation of a STACK that holds integers. Write a method public void CountPosNeg (Stack S) that takes Stack S as a parameter. After the call, the method prints the count of positive integers and negative integers on the stack. Note, the original must not be destroyed.

```
public void CountPosNeg (Stack S) {
```

Part c. [4 points] Write a recursive method to check whether the Binary Tree Node passed as a parameter is actually a Binary-Search-Tree. The method takes a Node (root of the Binary Tree) as a parameter and returns a boolean (true or false) value.

```
public boolean isBST(Node Root){
```

Part d. [4 points] Find the Asymptotic complexity (Big O) for the following algorithms by computing the number of primitive operations:

```
int k= 0;
for (i = 1; i <= n; i++) {
  for (j = 1; j <= 5; j++) {
    k = k + i + j;
  }
}
```

```
int k= 0;
for (i = 1; i <= n; i++) {
  for (j = i; j <= n; j++) {
    k = k + i
  }
}
```

```
public void print(int n){
  int k = n;
  for (int i = 1; i <= n; i++)
    for (int j = 1; j <= k; j++)
      System.out.println(j+i);
}

public void main(){
  Scanner Key= new Scanner(System.in);
  int n=Key.nextInt();
    for (i = 1; i <= n; i++)
      print(n);

}
```

```
for (i = 1; i <= n; i*=2) {
      System.out.println(i);
}
```

Question 4. [2 + 1 = 3 points]

Part a. [2 points] Given a hash table of size 9 that stores integers. Suppose that the hash function used is h(x) = x mod 9. Using the separate chaining method to resolve collisions, insert the following in this Hash table.

21, 4, 14, 12, 3, 18, 9

Show the hash table. How many collisions were observed?

Part b. [1 points] Would it be better to use a Hash-table with linear-probing? Why or why not?

-End of Exam

<Scratch sheet. DO NOT detach>