Prince Sultan University
CCIS - Department of Computer Science

Final Examination
Term 232

Course title: Data Structures and Algorithms **Course Code: CS 210**

Exam date: June 2, 2024 **Exam Time: 180** minutes

Name ____

Student ID									

Circle the name of your instructor and section time

Dr. Basit Qureshi **Dr. Syed Umar Amin**

8 AM **9 AM** **10 AM**

Instructions:

- This exam contains four questions with multiple parts.
- Time allowed: 180 minutes
- Closed Book, Closed Notes.
- Use of Calculators is allowed. Any electronic devices / smartphones etc. is strictly prohibited.
- Answer the problems on the exam sheets only. No additional attachments would be accepted.
- If you need extra space, use the back of a page.
- When the "time is over" is called, it is students' responsibility to submit his exam to the invigilator. Submitting completed exam 3 minutes after the "time is over" will incur a penalty.
- Do NOT use the erasable pens

Question No.	CLO	Max Score	Student's Score
Question 1:	1	15	
Question 2:	2	8	
Question 3:	3	8	
Question 4:	4	9	
Total (40)	-	40	

Question 1. [2 + 2 + 3 + 4 + 4 = 15 points] [CLO 1] **[/15 points]**

Part a. **(/2 points)**

Given the following statement, complete the table with the correct answer:

	Statement / Question	**Answer (in real numbers NOT Big-O)**
1	Let T be a binary search tree which has 13 levels. What is the maximum number of possible nodes in level 8.	
2	Let T be a binary tree which has 5 levels, what is the maximum number of nodes that tree T can have?	
3	You are given 99 values to insert a binary tree, i.e. you need 99 nodes to store the values. What is the minimum number of levels which you will need to store these values?	
4	You are given a binary search tree which has 78 nodes. The nodes were inserted in the tree in sorted order. How many levels do you expect the tree to have?	

Part b. **(/2 points)**

Your math teacher asks you to assist in representing an arithmetic expression in a binary tree. You are given the following arithmetic expression.

$$(((5+11) / (8 - 4)) * (4 + 2))$$

1. Draw the Binary tree that represents the expression.

2. Evaluate the value of the expression.

Part c. (/3 points)

Consider a priority queue implemented using a **max-heap**. Every node in the heap stores a key value pair **(Key, Value)** , where the key is the priority (integer) and value is a string.

1. Show the steps required to build a binary heap by inserting the following input: [2pt]

   (10,abd) (3,ijk) (2,xyz) (5,def) (15,abc) (9,pqr) (7,stu) and (4,stp).

2. Draw the heap resulting from **removing** the element of highest priority (KEY) from the heap obtained above. [1pt]

Part d. (/4 points)

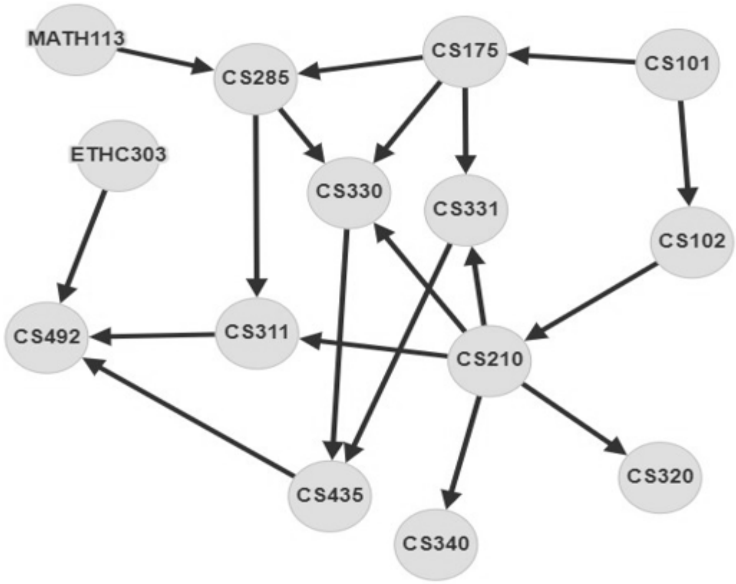
Sort the following integer array using Insertion sort and Selection sort. Compute how many Comparison operations and swap/exchange operations are conducted for each algorithm.

Selection Sort						Insertion Sort					
3	8	5	4	9	1	3	8	5	4	9	1
Number of Comparisons: Number of Swap/exchanges:						Number of Comparisons: Number of Swap/exchanges:					

Which algorithm is better?

Part e. (/4 points)

The following Digraph shows the courses an undergrad CS major student needs to take to graduate at Prince Sultan University. Give the **adjacency list** for the digraph showing the courses and prerequisites. [2pt]

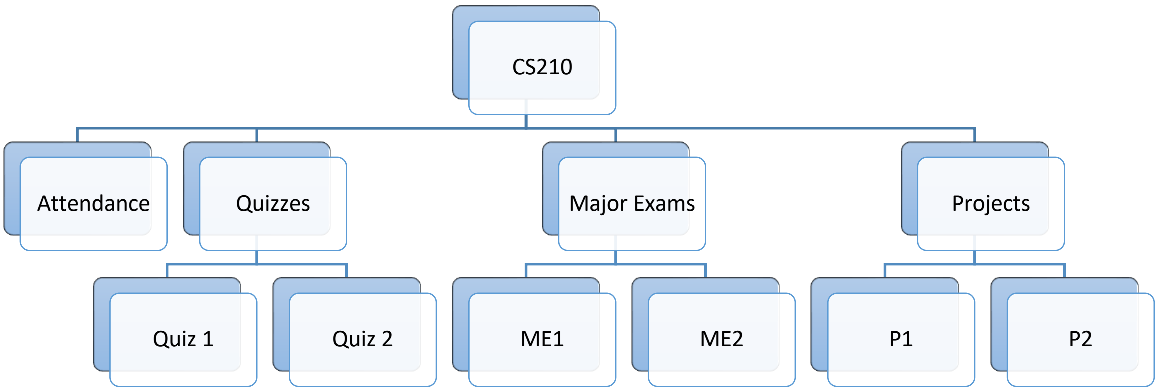


Provide the **Breadth First Search (BFS)** traversal of the graph starting at node CS101. [2pt]

Question 2. [3 + 3 + 2= 8 points] [CLO 2] **[/ 8 points]**

Part a. **(/3 points)**

Consider the requirements of this coursework (CS210) in the following tree:



Consider the following algorithm:

```
Algorithm CS210_CourseWork(v)
     for each child w of v
          CS210_CourseWork(w)
     Print(v)
```

1. **What does the above algorithm do?** [0.5pt]

2. **Draw the recursion trace for the algorithm** `CS210_CourseWork(CS210)` [2pt]

3. **What is the big-O notation of algorithm** `CS210_CourseWork`**?** [0.5pt]

Part b. (/ 3 points)

Give the Runtime as a Big-O notation for the following code snippets. Assume n ≥ 1.

<table>
<tr><th>Question</th><th>Big-O</th></tr>
<tr><td><pre>
void method1(int n){
        int x = 0;
        for (int i = 0; i < n; i++)
                for (int j = 0; j < n * n / 3; j++)
                        x += j;
}
</pre></td><td></td></tr>
<tr><td><pre>
boolean method2(int n){
        for(int i = 2; i < n; i++){
                if(n % i == 0)
                        return false;
        }
      return true;
}
</pre></td><td></td></tr>
<tr><td><pre>
void method3(int n){
        int sum = 0;
        for (int i = 1; i <= m; i++)//m is a constant
                for (int j = 1; j <= n; j*=2)
                        sum += (i+j);
}
</pre></td><td></td></tr>
<tr><td><pre>
void method4(int arr[], int F, int L, int K){
      int m = (F + L)/2;
      while (F <= L) {
            if (arr[m] < K)
                  F = mid + 1;
            else
                  {if (arr[m] == K)
                        {System.out.println("found at:" + mid);
                        break;}
                  else
                        L = m - 1;
                  }
            m = (F + L)/2;
      }
      if (F > L)
            System.out.println("not found!");
}
</pre></td><td></td></tr>
<tr><td><pre>
public int method5(int n){
      if (n <= 1)
            return 2;
      else
            return fun(n / 2) + fun(n / 2);
}
</pre></td><td></td></tr>
<tr><td><pre>
public int returnEnd (LinkedList S) {
      Node Temp = S.Head;
      while(Temp.next!=null)
            Temp = Temp.next;
      return Temp;
}
</pre></td><td></td></tr>
</table>

Part c. (/ 2 points)

Given the following statement, choose the correct answer:

	Statement / Question	**Answer (Big-O Notation)**
1	At any given time, the runtime to search for the maximum value in an AVL tree is: a) O(log n) b) O(1) c) O(n) d) O(n log n)	
2	The worst-case runtime for searching a key in a skewed Binary Search Tree is: a) O(log n) b) O(n) c) O(1) d) O(n log n)	
3	The space complexity needed to store **n** items in an AVL tree is: a) O(log n) b) O(n) c) O(1) d) O(n log n)	
4	The average-case runtime to delete the root of a Binary Search Tree is: a) O(log n) b) O(n) c) O(1) d) O(n log n)	

Question 3. [3 + 5 = 8 points] [CLO 3] **[/ 8 points]**

Part a. **(/ 3 points)**

Given the following input (12, 39, 70, 73, 66, 81, 65, 92) in a table with **size 10 and** the hash functions are:

f(x) = 8
g(x)= 10 - (x % 2);
h(x)=(x-5) % 2;
t(x) = (x^3) % 10

Choose the correct answer(s):

Question	Answer
If collisions are handled by separate chaining, the best hash function is: 1. f(x) 2. g(x) 3. h(x) 4. t(x)	
If collisions are handled by linear probing, the hash function with the highest # of probs is: 1. f(x) 2. g(x) 3. h(x) 4. t(x)	
Which hash functions will create only two chains if collisions are handled by separate chaining? 1. f(x) 2. g(x) 3. h(x) 4. t(x)	
If collisions are handled by separate chaining, the worst hash function is: 1. f(x) 2. g(x) 3. h(x) 4. t(x)	
Which hash function will hash all keys to the same index? 1. f(x) 2. g(x) 3. h(x) 4. t(x)	
Regardless of collision handling, which function would you choose if you could choose a hash function for your application? 1. f(x) 2. g(x) 3. h(x) 4. t(x)	

Part b. (/ 5 points)

Consider a hash-table T1 of size 10, storing integer keys that handles collision with double hashing.

$h(x) = x \bmod 10$
$d(x) = 7 - x \bmod 7$

Insert the following data in the hash table: 3, 15, 17, 22, 19, 16, 27, 35, 46, 8.
Compute the number of collisions and probes/displacement for each.

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Number of collisions with f(x)

____

Number of collisions with g(x)

____

Number of probes/displacements with f(x)

____

Number of probes/displacements with g(x)

____

Question 4. [4+ 3+ 2 = 9 points] [CLO 4] **[/9 points]**

Part a. **(/4 points)**

A Circular Linked List C is provided that stores an integer Key and integer Val in each node. Write a method that removes all instances of Val V from the list. The method returns a count/number of nodes removed.

```
+-------+    +-------+    +-------+    +-------+    +-------+    +-------+    +-------+
| Key:1 |    | Key:2 |    | Key:3 |    | Key:4 |    | Key:5 |    | Key:6 |    | Key:7 |
| Val:1 | -> | Val:1 | -> | Val:3 | -> | Val:4 | -> | Val:5 | -> | Val:3 | -> | Val:7 |
+-------+    +-------+    +-------+    +-------+    +-------+    +-------+    +-------+
     ^                                                                           |
     |---------------------------------------------------------------------------|
```

For example, after executing the method removeVal with V=3, the nodes with Key:3 and Key:6 should be removed.

```
public int removeVal(CircularLinkedList C, int V){
```

Part b. **(/3 points)**

You are required to process a series of tasks related to delivering packages for a shipment company. Initially, there are two stacks that store all the packages ‘Packages’ and returned packages ‘ReturnedPackages’, and there are two queues representing each delivery representative: ‘DeliveryMan1’ and ‘DeliveryMan2’ containing the packages to be delivered by the respective delivery man. The goal is to assign these packages to the two delivery men and keep track of the returned packages.

Scenario 1

Package: B
Package: C
Package: D

Stack ‘Packages’

Package: E

Stack “ReturnedPackages”

Package: A						

Queue ‘DeliveryMan1’

Queue ‘DeliveryMan2’

Scenario 2

Stack ‘Packages’

Package: E
Package: A

Stack “ReturnedPackages”

Package: D	Package: B					

Queue ‘DeliveryMan1’

Package: C						

Queue ‘DeliveryMan2’

Write the **sequence of operations** needed to achieve the final configuration in Scenario 2 for the provided stacks and queues.

Part c. (/2 points)

Write a method that returns the number of levels in a binary search tree. Your method will start a given node p and will count the levels recursively.

```
int countLevels(Node p) {
```

<This sheet is left blank intentionally. DO NOT detach>

<End of Exam>