Final Examination Cover Sheet

Course Code: CS210 **Course Title:** Data Structures and Algorithms **Term:** 231 **Year**: 2023 / 2024

Exam Duration: 3 hours **Exam Date:** 25-Dec-2023

Number of Exam Pages (including cover page): 13 pages (1 cover page + 12 questions pages)

Exam Rules

During the exam:

- Students are **NOT** allowed to **TALK** to each other.
- Students should keep the **bags / books / notes in the front of the exam hall.**
- Students are not allowed to **HAVE** or **USE CELL PHONE, SMART WATCHES, HEADPHONE or STUDY MATERIAL**
- . Students are ALLOWED to use **CALCULATORS**.

If you have any of the following items, give it to the invigilator right away before looking at the questions:

The PSU Honor Code

"أتعهد أنا الطالب الموقع أدناه بأنني قرأت التعليمات أعلاه وبأن أكون ملتزمًا بقوانين الجامعة."

"I have read the above guidelines and I do hereby pledge to conduct myself with honesty and integrity during the examination and abide by the rules and regulations of the University."

Student's Name:_____ **Student's Signature:** _____

Student's ID#:_____

CIRCLE YOUR SECTION TIME AND INSTRUCTOR NAME

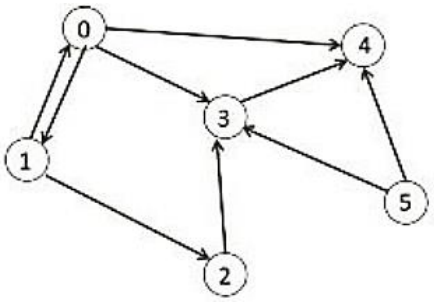
Dr. Basit Qureshi		Dr. Syed Umar Amin	Dr. Abdullah Alrajeh
8	9	10	11

Question No.	CLO	Max Score	Student's Score
Question 1:	**1**	**13**	
Question 2:	**2**	**10**	
Question 3:	**3**	**8**	
Question 4:	**4**	**9**	
Total (40)			

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

Part a. **(/ 4)**

Consider the following directed graph



1. Show the **adjacency matrix** for this graph. **(/ 1)**

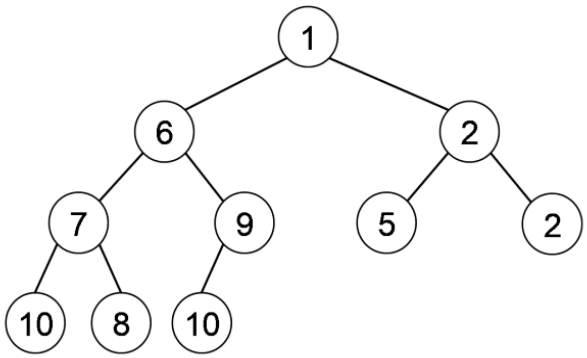
2. Show the **adjacency list** for this graph (for each linked list, order edges in increasing order of the vertex id). **(/ 1)**

3. Provide the **Depth First Search (DFS)** traversal of the graph starting at node 0. **(/ 1)**

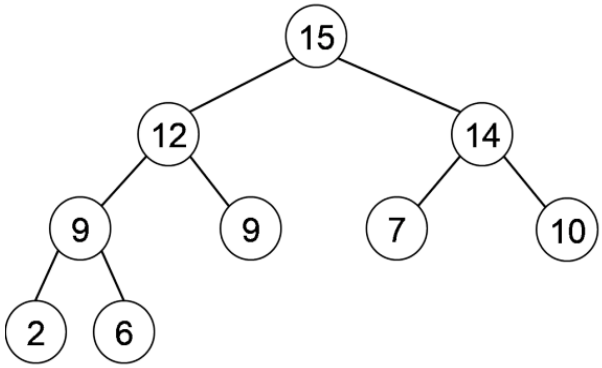
4. Provide the **Breadth First Search (BFS)** traversal of the graph starting at node 0. **(/ 1)**

Part b. **(/ 2)**

1. Draw the heap resulting from **inserting** an element with priority (KEY) 5 into the following **min-heap.** Show all swim (upheap) / sink (downheap) operations as needed.

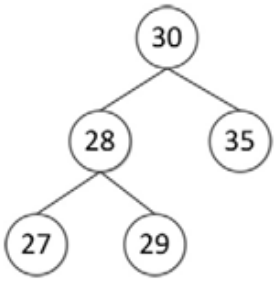


2. Draw the heap resulting from **removing** the element of highest priority (KEY) from the following **max-heap.** Show all swim (upheap) / sink (downheap) operations as needed.

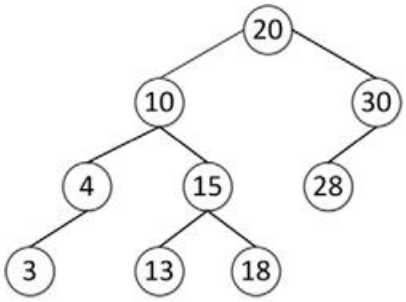


Part c. **(/ 2)**

1. For the AVL tree below, draw the resulting **AVL** tree obtained after **removing element 35.** Show the rotations needed to balance the tree.



2. For the AVL tree below, what is the result of the **AVL** tree obtained after **inserting element 17**? Show the rotations needed to balance the tree.



Part d. **(/ 2)**

A friend tells you that traversing a **Binary tree** resulted in:
In-order traversal: 7. 10. 13. 15. 19. 20. 22. and
Post-order traversal: 7. 13. 10. 19. 22. 20. 15

Draw the tree using the given information.

Part e. **(/ 3)**

Answer the following MCQs

1	Suppose you have numbers between 1 and 1000 in a binary search tree, and you want to search for the number 532. Which sequence could **not** be the sequence of nodes examined when searching for the number 532 in a binary search tree? There can be more than one. a. 750, 200, 400, 600, 450, 455, 500, 532. b. 925, 200, 910, 240, 920, 340, 350, 532. c. 935, 270, 340, 620, 290, 390, 350, 532. d. 800, 690, 680, 210, 260, 580, 510, 532.	
2	Which of the following array sequences would result in the Quicksort algorithm running in $O(n^2)$ time, assuming the pivot is the 1^{st} element? a. 10, 13, 17, 8, 16, 2, 15 b. 15, 2, 16, 8, 17, 13, 10 c. 16, 15, 17, 13, 10, 8, 2 d. 17, 16, 15, 13, 10, 8, 2	
3	The idea of in-place sorting is about efficiently utilizing space. Which of the following algorithms is **space in-efficient** when sorting an array with values: 201, 301, 101, 901, 501, 601. a. Quick sort b. Merge sort c. Insertion sort d. Selection sort	

Question 2. [3 + 7 = 10 points] [CLO 2] **[/ 10]**

Part a. **(/ 3)**

Give the Runtime as a Big Oh notation for the following code snippets. Assume $n \geq 1$.

	Runtime Big Oh

`int fun1(int n)`
`{`
`  if (n <= 0)`
`  return 1;`
`  else`
`  return 1 + fun1(n-1);`
`}`	

`int fun2(int n)`
`{`
`  if (n <= 0)`
`  return 1;`
`  else`
`  return 1 + fun2(n/2);`
`}`	

`void fun3(int n, int m, int o)`
`{`
`  if (n <= 0)`
`  {`
`  System.out.println(m+ " " + o);`
`  }`
`  else`
`  {`
`  fun3(n-1, m+1, o);`
`  fun3(n-1, m, o+1);`
`  }`
`}`	

`int fun4(int n)`
`{`
`  int m=0;`
`  for (int i = 0; i < n; i += 2) {`
`  m++;`
`  }`
`  if (n <= 0)`
`  return 1;`
`  else`
`  return 1 + fun4(n-1);`
`}`	

`int fun5()`
`{`
`  for (int i = 0; i < n; i++)`
`  {`
`  for (int j = 0; j < n * n; j++)`
`  {`
`  System.out.println("Hello");`
`  }`
`  }`
`}`	

`int fun6(AVLNode P, int i)`
`{`
`  if(P == null)`
`  return i;`
`  else`
`  return fun6(P.parent, i+1);`
`}`
`//Note: The method fun6 is called (P, 0), where P is any leaf node in an AVL tree.`	

Part b. **(/ 7)**

Provide correct answers for the following MCQs

The runtime for a depth-first search starts at vertex v for a directed graph with V vertices and E edges. a) O(V) b) O(E) c) O(V * E) d) O(V + E)	
The runtime for the quicksort algorithm applied to an array with all identical values a) O(n) b) $O(n^2)$ c) O(n log n) d) O(1)	
The runtime to search an edge in an adjacency Matrix for an undirected graph with V vertices and E edges. a) O(V) b) O(E) c) O(1) d) O(V + E)	
The worst-case scenario for inserting a key in a hash table using the linear probing method for collision avoidance a) O(1) b) O(log n) c) O(n) d) O(n log n)	
What is it called if several elements are competing for the same bucket in the hash table? a) Collision b) Diffusion c) Replication d) Duplication	
A hash function h(x) guarantees a uniform hashing. What is the runtime for inserting values in a hash table size 100? a) O(n) b) O(log n) c) O(n log n) d) O(1)	
In an AVL tree, what is the average-case runtime to search for a key at any given time? a) O(log n) b) O(1) c) O(n) d) O(n log n)	
The cost of inserting a key in a max-heap is: a) O(log n) b) O(1) c) O(n) d) O(n log n)	

The 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)	
Which of the following functions gives a linear runtime? a) $2^n + 100^{50}$ b) $2^{\log n} + 1000$ c) 1+2+ ... +(n-2)+(n-1)+n d) $2^{\log n}$+ n log n + 3 log n	
What is the best-case runtime complexity for removing the smallest value in a singly-linked list? a) O(log n) b) O(n) c) O(1) d) O(n log n)	
What is the runtime complexity for inserting a new element at the beginning of a singly-linked list? a) O(log n) b) O(n) c) O(1) d) O(n log n)	
Suppose that your application will have a huge number of *insert* operations, but only a few *remove* operations. Which data structure would you choose for the best runtime? a) Heap b) Unordered Array c) Singly Linked List d) Binary Search Trees	
Which of the following would be the fastest to find the largest value in a dataset using: a) Binary Tree b) Unordered circular linked list c) Max Heap d) AVL Tree	

Question 3. [5 + 3 = 8 points] [CLO 3] **[/ 8]**
Part a. **(/ 5)**

Sara wants to store some integers in a hash-table **T1 of <u>size 11</u>**. She wants to try out some hash functions to see if they would ensure uniform data distribution. The following functions are available f(x) and g(x).

$$\boldsymbol{f(x) = (x \,\%\, 5) * 2}\,, \qquad \boldsymbol{g(x) = (x * 2) \,\%\, 11}$$

Insert the following data in the hash table: 3, 15, 17, 22, 19, 16, 27, 35, 46, 8, 11.
Assume that **collisions** are resolved using **the linear probing** method; compute the number of collisions and probes/displacement for each.

	f(x)
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

	g(x)
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

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)

_____

Comparing the Hash-function f(x) and g(x). If you were going to choose a hash function in your application, which function would you go with? Explain why?

Part b. **(/ 3)**

Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) in a table with **size 10** and the hash functions are:

h(x)=x % 2;
g(x)= x % 11;
f(x) = 1;
t(x) = 10 - (x % 1);

Which of the following statements is true for h(x)? a) 9679, 1989, 4199 hash to the same index. b) 4322, 1334, 6173 hash to the same index. c) All elements hash to the same index. d) Each element hashes to a different index.	
Which of the following statements is true for g(x)? a) 9679, 1989, 4199 hash to the different index. b) 4322, 1334, 6173 hash to the same index. c) Inserting 4199 gives a collision. d) Hash function ensures uniform distribution.	
Which of the following statements is true for f(x)? a) 9679, 1989, 4199 hash to the different index. b) 4322, 1334, 6173 hash to the same index. c) Inserting 4199 does not give a collision. d) All values hash to the same index.	
Which hash function gives the worst indexing performance a) h(x) b) g(x) c) t(x) and f(x) d) t(x) and h(x)	
If the collisions are handled by separate chaining, which hash function would result in least number of collisions: a) h(x) b) g(x) c) t(x) and f(x) d) t(x) and h(x)	
Which technique avoids the clustering problem in Hash tables a) Linear Probing b) Double Hashing c) Collision Handling d) Uniform Hashing	

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

Part a. **(/ 3)**

You are required to process a series of tasks related to books in a library. Initially, there are two stacks: **'Fiction' (F)** and **'NonFiction,' (NF)** containing books from the respective genres. The goal is to organize these books into two separate shelves and maintain a queue, **'RecentlyReturned,' (RR)** to keep track of books returned to the library. **You are not allowed to use any additional data structures.**

Initial State:

Lord of the Rings
Dune,
Hunger Games
Harry Potter

Stack F

AstrophysicsforYoungPeople
Freakonomics
Cosmos
Sapiens

Stack NF

Queue RR

Final State: After Organization

Lord of the Rings
AstrophysicsforYoungPeople

Stack F

Dune
Freakonomics
Cosmos
Sapiens

Stack NF

Hunger Games	Harry Potter					

Queue RR

Write the sequence of operations needed to achieve the final configuration of the **'Fiction,' 'NonFiction,'** and **'RecentlyReturned'** structures.

Part b. **(/ 3)**

Write a method called **removeBack** that takes an integer k as a parameter. It **removes k number of nodes from the end of a singly linked list**. Method removeBack is part of the List class.

For example,
If a list L stores this sequence of values: [8, 17, 9, 24, 42, 3, 8] a call L.removeBack(4), would result in [8, 17, 9]
Assume that if k is greater than or equal to the length of L, removeBack will remove all nodes in L.

```
void removeBack(int k)
{
```

Part c. **(/ 3)**

Assume that an array A stores a BinaryTree of integer values. Write a method Ancestors(int [] A, int p) that returns all the ancestors of a given node at position p. The method returns a string value.

Example: For the following array, a call Ancestors(A, 12) would return “16, 6, 10”

0	1	2	3	4	5	6	7	8	9	10	11	12
10	3	6		19	16				7	8		20

string Ancestors(int [ ] A, int p)

{

<End of Exam>