

A

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Final
Duration: 1 Hour 30
Minutes
Number of
Questions: 5

CSE220: Data Structures

Semester: Spring 2024
Full Marks: 45
No. of Pages: 4

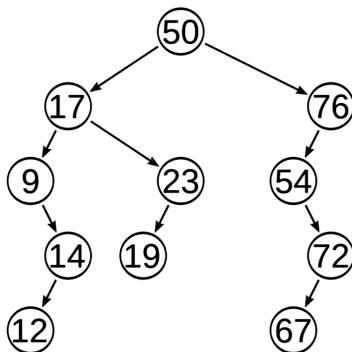
Name: (Please write in CAPITAL LETTERS)	ID:	Section:
--	-----	----------

- ✓ Answer all 5 questions. No washroom breaks. Understanding questions is part of Exam.
- ✓ At the end of the exam, put the question paper inside the answer script and return both.

Question 1: CO1 [5 Points]

Write the correct answer in your answer script:

Check the BST below:



I. Which is an unbalanced node in the above tree?

- a. 50, b. 23, c. 54, d. 67

II. Which is the predecessor of Node 17 in the above BST?

- a. 23, b. 9, c. 50, d. 14

III. Which of the following linear array is the most accurate and ideal representation? (N indicates None value)

- a. [1, 2, N, N, 4, 5]
b. [1, 4, -3, N, 5, 7]
c. [1, -2, 3, N, N, N]
d. [N, N, 1, 4, 5, 7]

IV. Heap = [N, 19, 15, 12, 13, 7, 5, 1, -1, 3]

This array indicates a maxHeap. Value 20 is inserted in the heap. What will be the index of the value 7 in the new heap?

- a. 4, b. 5, c. 9, d. 10

V. Which of the following is FALSE?

- a. Linked List does not allow random access.
- b. Head node cannot be directly accessed from the Tail node in a Singly Circular Linked List.
- c. Each node has two pointers in a Doubly Circular Linked List.
- d. Tail node can be directly accessed from the Head node in a Doubly Circular Linked List.

Question 2: CO3 [10 Points]

You have an interesting hash function for a hash table. The hash-function takes a singly linked list where node values are characters and the characters are put in the list using some unknown permutation. The hash function takes a string key as input and the head of the linked list. Then the hash for the key is the mod by 10 of the sum of the indexes of the characters of the list in the string. Implement this hash function. If any node value is not present in the string, then consider its index -1. If any character of the string occurs multiple times, then consider the index value of the last occurrence. **You can not use any built-in function except range. Consider the Node class has elem and next variable. No need to write the Node class. You can not use any data structures other than the given data structures.**

Python Notation:

```
def hash(list, st):  
    # To Do
```

Sample Input	Sample Output	Explanation
hash(list, st) list = D → A → T → A st = "SADA"	7	Index of D is 2 and A is 3. T is not present in the string hence -1. $(2 + 3 - 1 + 3) \% 10 = 7$

Question 3: CO4 [10 Points]

Given a Binary Tree, complete a function that finds if the given Binary Tree is a SumTree or not. A SumTree is a Binary Tree where the value of a node is equal to the sum of the nodes present in its left subtree and right subtree. An empty tree is SumTree and the sum of an empty tree can be considered as 0. A leaf node is also considered as SumTree. The function should return True if it is a SumTree and otherwise False.

Consider the Node class for Binary Tree already defined with elem, left and right variables. You can use helper functions.

YOU CANNOT USE LIST OR DICTIONARY, any built-in function, or global variables.

Python Notation:

```
def sumTree(root):  
    // To do
```

Here root refers to the tree below.

Sample Input	Sample Output	Explanation
<pre> 26 / \ 10 3 / \ \ 4 6 3 </pre>	True	<p>For 26 Sum of left(20)+sum of right(6)=26</p> <p>For 10 Sum of left(4)+Sum of right(6)=10</p> <p>For 3 Sum of left(0)+Sum of right(3)=3</p>

Question – 4: CO3 [10 Points]

You are given an array A of size N. You can perform an operation in which you will remove the largest and the smallest elements from the array and add their difference back into the array. You are given an integer k that will define the number of times you have to perform the operation. Write a method heapSum() that takes an array, an integer k as parameters, performs k operations on the array, and returns the sum of the array elements after the operations.

You are given MaxHeap and MinHeap classes. The methods in the MaxHeap class are - MaxInsert(), MaxDelete() and the methods in the MinHeap class are - MinInsert(), MinDelete(). The constructor of both classes does not take any parameters. The method works according to their names and no need to implement any of the methods.

YOU MUST USE THE GIVEN HEAP DATA STRUCTURES TO SOLVE THIS PROBLEM.

Hint: No need to modify the given array. Remember that the underlying data structure of a heap is an array.

Python Notation:
def heapSum(A, k):
 # To Do

Sample Input	Sample Output	Explanation
<pre> heapSum(A, k) A=[3, 2, 1, 5, 4,] k = 2 </pre>	9	<p>After 1st operation, the array will become, A = [3,2,4,4]. After the 2nd operation, the array will become, A = [3,2,4]. sum of elements = 9.</p> <p>Here the shown value of A is just to explain the output and does not mean the state of the given array after each operation</p>

Question – 5: CO2 [10 Points]

Suppose you have a set of 10 vertices which are denoted as A, B, C, D, E, F, G, H, I, J.
The matrix representation below shows the relationship between any of these two vertices.

	A (0)	B (1)	C (2)	D (3)	E (4)	F (5)	G (6)	H (7)	I (8)	J (9)
A (0)		5	3							
B (1)				7						
C (2)				4	6					
D (3)					2					
E (4)										
F (5)							8			
G (6)								9		
H (7)									10	
I (8)										
J (9)										11

[Note: Vertex A is mapped as index position 0, Vertex B is mapped as index position 1 in the matrix and so on]

- Using the above matrix representation, draw an adjacency list representation of the above nodes. [3]
- Determine** whether the graph is directed or undirected. If it is undirected, transform it into a directed graph and if it is directed, transform it into an undirected one using the above matrix representation. Write down the resulting matrix in your script [1+2]
- Suppose that the vertices A to J have the following weights 31, 19, 25, 40, 45, 35, 33, 27, 14, 21, 37 respectively.
 - Draw a Binary Search Tree using the vertices A to J sequentially based on their weights. Write down the node's weights while drawing the BST. [2]
 - Delete the node with the key of value 31 with the help of its predecessor of the above tree and draw the new tree. [2]