International Islamic University Chittagong

Department of Computer Science and Engineering

B. Sc. in CSE Semester Final Examination, Autumn 2023

Course Code: CSE 2321 Course Title: Data Structures

Time: 2 hours 30 minutes *Total marks: 50*

[Answer the following questions. Figures in the right hand margin indicate full marks.]

CO DL

Group A

[Answer the following questions]

1. a) Write the two properties of a recursion function. Find the base criteria of the following recursive functions. 3 CO4 C5

i. $H(N)=\begin{cases}3*N & IF\ N<5\\ 2*H(N-5)+7 & Otherwise\end{cases}$

ii. $F(N)=\begin{cases}1 & IF\ N=0\\ N.F(N-1 & IF\ N>0\end{cases}$

iii. $GCD(A,B)=\begin{cases}GCD(B,A) & IF\ A<B\\ A & IF\ B=0\\ GCD(B,MOD(A,B)) & Otherwise\end{cases}$

b) What do you mean by Tower of Hanoi problem? Draw a schematic diagram to solve the Tower of Hanoi problem for n=4 disk. 5 CO4 C5

OR

b) Find the value of factorial five (6!) using recursion. Show each step and also show the status of stack used in the recursion. 5 CO4 C5

c) The value of REAR is increased by 1 in case of insertion in queue and the value of FRONT is increased by 1 in case if deletion from queue. Do you agree with this statement and why? 2 CO4 C5

2. a) What is *linked list*? What are the advantages and disadvantages of linked list over the linear array? 2 CO1 C1

OR

a) What is *circular header list*? Write the advantages of circular header list over ordinary linked list. 2 CO1 C1

b) You are given a linked list containing an odd number of elements. You don't know how many numbers there are in that list. Write an algorithm or C/C++ code to find the *middle element* of this list. 4 CO3 C2

OR

b) You are given a two-way linked list containing an odd number of elements. You don't know how many numbers there are in that list. Write an algorithm or C/C++ code to find the middle element of this list by iterating only once over the list. 4 CO3 C2

c) Consider the *alphabetized* list of patients in Fig-2c. Determine the changes in the data structure if i) Ford is added to the list and then ii) Lane is deleted from the list. 2 CO1 C2

START 5 AVAIL 10

	1	2	3	4	5	6	7	8	9	10	11	12
BED	Kirk		Dean	Max	Adam		Lane	Green	Samu		Fields	Nelso
Link	7	6	11	12	3	0	4	1	0	2	8	9

Fig-2c

d) A **DEQUE** is a data structure that allows insertion and deletion of elements from both ends that allows the following four operations on it: 2 CO1 C3

1. **PUSH_FRONT(X):** This function adds an item X in front of its elements.
2. **PUSH_BACK(X):** This function adds an item X at the end of its elements.
3. **POP_FRONT():** This function deletes the first element (if there is any).
4. **POP_BACK():** This function deletes the last element (if there is any).

Suppose you are building a **DEQUE** where the underlying data structure would be a one way linked list. What would be the *time complexity* of these four operations? Explain.

Group B

[Answer the following questions]

3. a)	What do you mean by the complexity of algorithms? Analysis the complexity of selection sort algorithm.	5	CO3	C2
b)	Suppose the following characters are stored in an array A: **M, U, S, E, U,M** Apply ***selection sort*** algorithm to sort the array A and show each pass separately.	5	CO3	C2
4. a)	Analyze the differences between Binary tree and Complete binary tree with figure. Construct an *expression tree* **T** for the following arithmetic expression: **(A + B * C) + ((D / E - F) * G)** Traverse the tree T in **preorder, postorder** and **inorder.**	5	CO5	C2
b)	Suppose the following eight numbers are inserted in order into an empty binary search tree T: 50, 33, 44, 22, 77, 35, 60, 40 Draw the tree T and show each step.	4	CO1	C2
	OR			
b)	What do you mean by *max heap*? Build a max heap from the following list of numbers: **23, 29, 49, 21, 57, 62, 73, 54**	4	CO1	C2
c)	Explain which data structure is most efficient to find the *top 10 largest items* out of *1 million items* stored in file? Why? i) Min heap ii) Max heap iii) BST iv) Sorted array	1	CO4	C4
5. a)	What is a *graph*? Consider the following graph G in Fig-5a:	2.5	CO1	C2

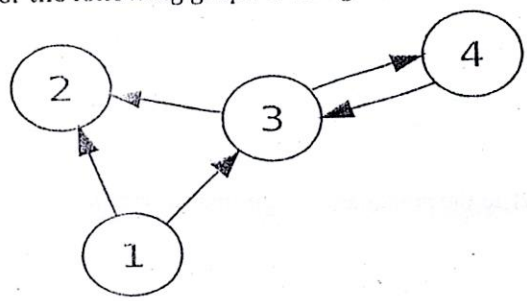

Fig-5a

i) Find the *adjacency matrix* A of the graph **G**.

ii) Find the *adjacency list* of the graph **G**.

b)	Use *Warshall's algorithm* to find the *path matrix* for the graph **G** in Fig **5a**. Show each step.	3.5	CO1	C2
	OR			
b)	Find the *path matrix* for the graph **G** in Fig **5a** using powers of the adjacency matrix **A**.	3.5	CO1	C2
c)	Traverse the graph G shown below in **breadth first order, depth first order** and construct the **breadth first** and **depth first spanning trees**. Start from node **A** [if last digit of your ID is *odd*] / **E** [if last digit of your ID is *even*].	4	CO1	C2