INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SRI CITY

TERM-II EXAMINATION – SPRING 2024

Data Structures and Algorithms

CSE: UG1 (PC) **Date: 16-03-2024**

Duration: 90 Minutes (03:30-05:00PM) **Max. Marks: 30**

Instructions: **Roll No:** S20230010185

1. This is a closed book exam. No textbooks, notes, paper or electronic devices are allowed.
2. All questions are compulsory. Clearly explain the step-by-step answer for each question.
3. Please attach the question paper along with the answer sheet during submission.

Section-A

1.	Which of the following statements is/are true about circular linked lists? i. Circular lists are complex to implement as compared to singly linked lists. ii. Reversing of circular lists is complex as compared to doubly linked lists. iii. If circular linked lists are not traversed carefully, then we could end up in an infinite loop. iv. Entire circular linked list can be traversed from any node. a. i, ii and iii b. i, ii and iv c. i and ii d. All of the above	[1 Mark]
2.	In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is a. log n b. n/2 c. log n-1 d. n	[1 Mark]
3.	Which of the following techniques stores data in a separate entity in case of a collision? a. Open addressing b. Chaining using linked list c. Linear probing d. Double hashing	[1 Mark]
4.	What is the worst case time complexity for the search function in a hash table using list head? a. $O(n)$ b. $O(n\,logn)$ c. $O(1)$ d. $O(logn)$	[1 Mark]
5.	Suppose a circular queue of capacity (n – 1) elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = -1. The conditions to detect queue full and queue empty are a. Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT b. Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR c. Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT d. Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT	[1 Mark]

6.	Suppose you are given an implementation of a queue (Q) of integers.
	Consider the following function:
	void f (queue Q) {
	int i ;
	if (!isEmpty(Q)) {
	i = dequeue(Q);
	f(Q);
	enqueue(Q, i);
	}
	}
	What operation is performed by the above function f on queue Q ?
	a. Leaves the queue Q unchanged
	b. Reverses the order of the elements in the queue Q
	c. Deletes the element at the front of the queue Q and inserts it at the rear keeping the other elements in the same order
	d. Empties the queue
		[1 Mark]

7.	If the sequence of operations, push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2), pop are performed on the stack, the sequence of popped elements are a. 2,2,1,1,2 b. 2,2,1,2,2 c. 2,1,2,2,1 d. 2,1,2,2,2	[1 Mark]
8.	The following postfix expression with single digit operands is evaluated using stack 8 2 3 ^ / 2 3 * + 5 1 * The top two elements of the stack after the 1st * is evaluated are a. 6, 1 b. 5, 7 c. 3, 2 d. 1, 5	[1 Mark]
9	Binary Search can be categorized into which of the following? a. Brute Force technique b. Divide and conquer c. Greedy algorithm d. Dynamic programming	[1 Mark]
10	What is the worst case complexity of binary search using recursion? a. $O(n\, logn)$ b. $O(logn)$ c. $O(n)$ d. $O(n^2)$	[1 Mark]

Section-B

11.	Convert the below infix expression to prefix using stack. Also write how it is more efficient than normal conversion without using a stack. K + L - M*N + (O^P) * W/U/V * T + Q	[4 Marks]
12.	Consider a circular queue with size 6 and has 6 elements 10, 20, 30, 40, 50 and 60 where Front = 0 and Rear = 5. Perform the following operations and display the Front and Rear values after every operation. enqueue(70), dequeue(), enqueue(10), dequeue(), enqueue(20), dequeue(), display().	[4 Marks]
13.	Consider the following elements K (9,10,11,19,12,18,48). Insert the elements into the hash table of size 9 using linear and quadratic probing. Compare the total number of collision occurred [Use H(K)= element % table size]	[4 Marks]
14.	You are given an array of n distinct integers $x_1, x_2, x_3, \dots, x_{n-1}$. You are also	[4 Marks]

	given an integer t. Your task is to find out whether $t = x_i$ for $0 \le i < n - 1$. Propose an algorithm to solve this problem. Your algorithm must have an expected running time of $O(n)$.	
15.	a. Give a suitable **C structure** to represent a doubly linked list of characters.	[2 Marks]
	b. State the output (return value) of the function **FindSpecialChar(str)** for the input strings, **str = "merge", "memory", and "recursion"**. Explain what the **FindSpecialChar(str)**is returning for each string.	[2 Marks]

```
char FindSpecialChar(char *str) {
    Node *head = (Node *)malloc(sizeof(Node));
    Node *p = head;
    Node *q = 0, *r = 0;
    int i, n = strlen(str);

    // Form a list with the characters of the string
    for(i = 0; i < n - 1; ++i) {
        p->data = str[i];
        p->next = (Node *)malloc(sizeof(Node));
        p = p->next;
        p->next = 0;
    }
    p->data = str[n - 1];

    // Find a special character
    q = p = head;
    while (p) {
        p = p->next;
        if (p) {
      p = p->next;
      q = q->next; }
     }
   return q->data;
```