Total No. of Questions—12] [Total No. of Printed Pages—7

[3662]-204

S.E. (Computer Engineering) (I Sem.) EXAMINATION, 2009

DATA STRUCTURES AND ALGORITHMS

(2008 PATTERN)

Time : Three Hours **Maximum Marks : 100**

N.B. :— (*i*) Answer *three* questions from Section I and *three* questions from Section II.

(*ii*) Answers to the two Sections should be written in separate answer-books.

(*iii*) Neat diagrams must be drawn wherever necessary.

(*iv*) Figures to the right indicate full marks.

(*v*) Assume suitable data, if necessary.

SECTION I

1. (*a*) Write the following statements in 'C' : [4]

(*i*) print the address of a float variable 'P'.

(*ii*) declare and initialize an array to four characters, *a, b, c, d*.

(*iii*) declare a pointer to a function *'f'* which accepts integer as parameter and does not return anything.

(*iv*) read a string which will contain only alphabets.

P.T.O.

(*b*) Write a 'C' program using pointers and functions to :

(*i*) Read a string

(*ii*) Append a string to another string

(*iii*) Copy a string to another string.

without using library functions. [6]

(*c*) Write iterative and recursive 'C' functions to find the greatest common divisor of two integers. [6]

Or

2. (*a*) What is pointer in 'C' ? Write the output of the following code : [6]

```
main( )
{
    static int a[ ] = {10, 20, 30, 40, 50};
    int * ptr = a;
    static int arr[2][2] = {1, 2, 3, 4};
    char str[ ] = "ABCD × 4#";
    char * s = str + 2;
    int i, j;
    for(i = 0; i < 5, i++)
        printf("%d", * ptr++);
    for(i = 0; i < 2; i++)
        for(j = 0; j < 2; j++)
        printf("%d\,n", * (*(n + i) + j));
    printf("%c\n%c\n%c\n", *(str + 2), *s++, *--s);
}
```

(*b*) Write a 'C' program to count the number of special characters and lines in a text file and append it to same file. [6]

(*c*) Differentiate between the following : [4]

(*i*) Call by value and call by reference

(*ii*) Structure and union.

3. (*a*) Define the following terms : [4]

(*i*) Data structure

(*ii*) Pseudocode

(*iii*) Abstract data type

(*iv*) Big O notation.

(*b*) Write an algorithm to find the sum of integers in an array of size 'n' and explain the algorithm components with respect to your algorithm. [6]

(*c*) Explain the classification of datastructures with an example each. [6]

Or

4. (*a*) Give the basic statement constructs in an algorithm with example. [4]

(*b*) Write iterative algorithm to find the factorial of a number. Determine the efficiency of your algorithm. [6]

(c) Find the frequency count of each statement and determine the time complexity of the code given below : [6]

```
printf("%d", m);
printf("%d", n);
for(i = 0; i < = m; i++)
    for(j = 0; j < = n; j++)
        {
            count++;
            a + = b[count];
        }
```

5. (a) Write polynomial as ADT. Represent the following polynomials using array : [6]

(i) $5x^8 - 2x^6 + 2x^4 + 10$

(ii) $7x^5 - 2x^3 - 7x^2 - 2$.

(b) Write pseudo 'C' algorithm to find the fast transpose of a sparse matrix. Analyse the time complexity of the algorithm. [6]

(c) Derive the address calculation formula for multidimensional array, $A[u_0]\ [u_1]\ [u_2]\ \ldots\ldots\ [u_n]$ in row major and column major representation. [6]

Or

6. (*a*) Justify array as a linear data structure. Write ADT for an array. [6]

(*b*) Write pseudo 'C' algorithm to add two polynomials represented using array and analyze the time complexity. [6]

(*c*) Explain the representation of a sparse matrix with an example and state its advantages. [6]

SECTION II

7. (*a*) Give any *two* applications of sorting and searching. [4]

(*b*) Write pseudo 'C' algorithm for merge sort and determine its time complexity. [6]

(*c*) Write pseudo 'C' algorithm to search an element using binary search and analyze its time complexity for best and worst cases. [6]

Or

8. (*a*) Write pseudo 'C' algorithm for insertion sort and determine its time complexity. [6]

(*b*) Write the contents of list and each bucket, after each pass using radix sort for the following list of numbers :

10, 2, 15, 246, 37, 4, 25, 62, 100, 17. [6]

(*c*) What is sort efficiency and sort stability ? [4]

9. (*a*) What do you mean by dynamic memory allocation ? How is it done in 'C' ? [4]

(*b*) Write 'C' functions for create, insert at end, delete a given node and display list, operations on a singly linked list. [6]

(*c*) Write a 'C' function to concatenate two circular linked lists. [6]

Or

10. (*a*) Represent the following polynomials using circular linked list : [4]

(*i*) $12x^6 - 2x^5 - 3x + 10$

(*ii*) $3x^7 + 5x^2 + 2x$.

(*b*) What is doubly linked list ? Explain any *two* applications of doubly linked list. [6]

(*c*) Write 'C' functions for the following operations by choosing appropriate linked list : [6]

(*i*) to search a given element in list

(*ii*) to traverse the list from any node.

11. (*a*) Define Stack. Write algorithms for operations on a stack represented using array. [6]

(*b*) Write pseudo 'C' algorithm to convert infix expression to postfix expression using linked stack. [6]

(*c*) Write a 'C' function to reverse a given string using stack. Check a given string is palindrome or not, using this function. [6]

Or

12. (*a*) Define queue. Write linear queue as ADT using linked organization. [6]

(*b*) How the circular queue solves the problem in linear queue ? Give the representation of circular queue using array and linked list. [6]

(*c*) Write short notes on the following : [6]

(*i*) Queue Simulation

(*ii*) Priority queue.