

Total No. of Questions : 4]

SEAT No. :

P5189

[Total No. of Pages : 2

[6188]-141

B.E. (Computer Engineering) (Insem)
DESIGN AND ANALYSIS OF ALGORITHMS
(2019 Pattern) (Semester-VII) (410241)

Time : 1 Hour]

[Max. Marks : 30

Instructions to the candidates:

- 1) Answer the questions Q.1 or Q.2; Q.3 or Q.4.
- 2) Neat diagrams must be drawn wherever necessary.
- 3) Figures to the right indicate full marks.
- 4) Assume suitable data, if necessary.

- Q1)** a) Given the fastest computer and hypothetically infinite memory, do we still need to study algorithms? Justify. [2]
- b) How can we related algorithms to technology? Briefly explain. [6]
- c) Consider an array A of n integers which are already in sorted order. Let x be the number being searched in the array A in a liner fashion. The code fragment performing this task is given below: [7]

```
int lin _ search (int A [])
{
    i=0; flag=0;
    do { if (x == A [i]) then
        return (1); //Number found
    else
        i++;
    } while (i<n);
    return (0); // Number not found.
}
```

- i) Is this code fragment efficient? (We wish to use linear search only). Justify your answer.
- ii) Does it attribute to any design issue with respect to iterative algorithm? Briefly explain.

OR

P.T.O.

Q2) a) What is iterative algorithm? Explain iterative algorithm design issues using suitable examples. [8]

b) Consider the following algorithm to find the square of a number:

```
int sqr(int n)
{
    if (n == 0) return 0;
    else return (2n+sqr(n-1)-1)
```

Prove the correctness of this algorithm using principle of mathematical induction or otherwise. [7]

Q3) a) Briefly explain P and NP problems in the context of complexity theory. Give suitable example. [8]

b) If $f(n) = O(g(n))$ then does it imply $g(n) = O(f(n))$? Discuss. [5]

c) Comment on the statement “Best case analysis of algorithm may not give clear idea of performance” [2]

OR

Q4) a) What is SAT AND 3-SAT problem? Prove that 3-SAT problem is NP complete. [8]

b) What do you understand by best case, worst case and average-case behaviour of an algorithm? Is an average case efficiency an average of best-case, worst-case efficiencies? Justify answer. [7]

