(b) Design the algorithm of insertion sort for sorting number in decreasing order.

(c) Arrange the following function f(n), g(n) and h(n) into increasing order of their asymptotic growth :

$$f(n)=4^{(n*\lg n)}+n^2+6$$

$$g(n)=n^3+\lg(n!)+n$$

$$h(n)=\lg(n^n)+8^{(\lg n)}+n$$

H Roll No. .....

TCS–502/TIT–502

B. TECH. (CS/IT) (FIFTH SEMESTER)
MID SEMESTER EXAMINATION, 2019

DESIGN AND ANALYSIS OF ALGORITHMS

Time : 1 : 30 Hours

Maximum Marks : 50

Note : (i) This question paper contains two Sections.

(ii) Both Sections are compulsory.

Section—A

1. Fill in the blanks/True-False : (1×5=5 Marks)

(a) Time complexity of Merge sort is given by $\Theta(....)$.

(b) If $T(n)=36*T(n/6)+n*n*\lg n$, then $T(n)=\Theta(....)$.

(c) If $T(n)=n^3+\lg n^n+2^{\lg n}$, then $T(n)=\Theta(.....)$.

(d) Best case of Heap Sort when element of array is in decreasing order. (True/False)

(e) Quick is better than merge sort in most of the cases according to time. (True/False)

2. Attempt any *five* parts : (3×5=15 Marks)
   (a) Define term Algorithms with its property.
   (b) Give the Recurrence relation for Quick Sort in worst case and solve it.
   (c) Differentiate between divide and conquer and dynamic programming.
   (d) Design algorithm for Brute Force string matching for multiple occurrence.
   (e) Design algorithm for fractional knapsack problem.
   (f) Derive runtime complexity of Merge function used in Merge Sort.

Section—B

3. Attempt any *two* parts of choice from (a), (b) and (c). (5×2=10 Marks)
   (a) Solve $T(n) = n * T(n-1) + 1$ if $n > 1$ and $T(n) = 1$ if $n = 1$.
   (b) Find out runtime complexity of the following code (assume all variables as integer) :

```
for(i = n; i >= 1; i = i/2)
{for(j=1; j<=n; j++)
{p = q + r;}
}
```


   (c) Explain Recursive Tree method with the help of example.

4. Attempt any *two* parts of choice from (a), (b) and (c). (5×2=10 Marks)
   (a) Solve the following recurrence relation using Master's method :
      (i) $T(n)=4*T(n/2) + n * \lg n$
      (ii) $T(n)=24T(n/4) + n^2$
   (b) Design the algorithm of selection sort for sorting numbers in the decreasing order and derive the Time Complexity for Worst Case.
   (c) Give solution for the following fractional-Knapsack problem (Knapsack Size=32) :

Item	Cost	Weight
1	150	15
2	100	18
3	100	12
4	150	8

5. Attempt any *two* parts of choice from (a), (b) and (c). (5×2=10 Marks)
   (a) Apply heap sort on the following sequence to sort and show intermediate steps :

32, 4, 11, 9, 5, 7, 3, 16