JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -1 EXAMINATION- 2024

B.Tech-I Semester (CSE/IT/ECE/CE)

COURSE CODE (CREDITS): 18B11CI412 (3) MAX. MARKS: 15

COURSE NAME: Design & Analysis of Algorithms

COURSE INSTRUCTORS: YGL. AKJ, ARV, DHA MAX. TIME: 1 Hour

Note: (a) All questions are compulsory.

(b)Marks are indicated against each question in square brackets.

(c) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

Q.1 [CO1] (1+1+3)

a) Solve the relation $T(n) = 2T[n/ 2] + n \log n$ using Master Theorem.

b) Solve the relation $T(n) = 7T(n/2) + \Theta(n 2)$ using recursion method.

c) Design the recursive function for the given code and also illustrate the running time analysis in terms of growth of functions.

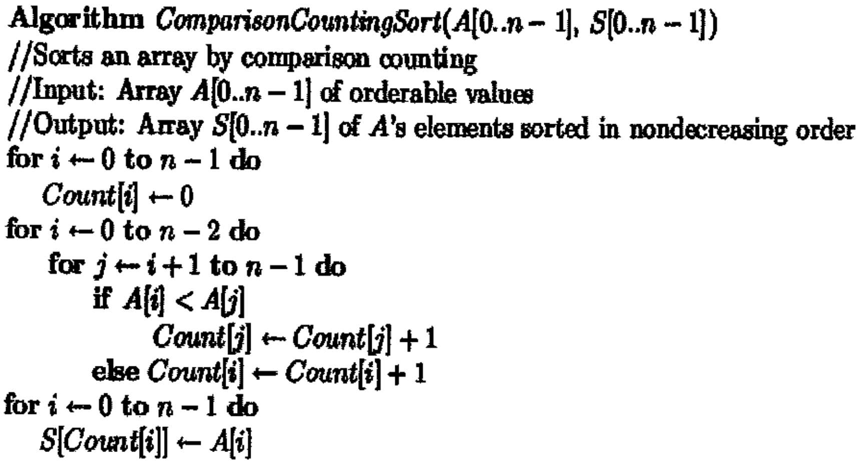
```
int weird(int A[], int N)
{
if (N<=4)
 return 100;
 if (N%5==0)
return weird(A,N/5);
 else
return weird(A,N-4)+weird(A, N-4);
}
```

Q.2 [CO1] Consider the below mentioned algorithm for the sorting problem that sorts an array by counting, for each of its elements, the number of smaller elements and then uses this information to put the element in its appropriate position in the sorted array. (3+1+1)

a) Apply this algorithm to sorting the list 60, 35, 81, 98, 14, 47.

b) Is this algorithm stable? Justify your answer.

c) Is it in place? Justification needed.



```
Algorithm ComparisonCountingSort(A[0..n − 1], S[0..n − 1])
//Sorts an array by comparison counting
//Input: Array A[0..n − 1] of orderable values
//Output: Array S[0..n − 1] of A's elements sorted in nondecreasing order
for i ← 0 to n − 1 do
    Count[i] ← 0
for i ← 0 to n − 2 do
    for j ← i + 1 to n − 1 do
        if A[i] < A[j]
            Count[j] ← Count[j] + 1
        else Count[i] ← Count[i] + 1
for i ← 0 to n − 1 do
    S[Count[i]] ← A[i]
```

Q.3 [CO1] Design a brute-force algorithm for computing the value of a polynomial at a given point (x_0) and determine its worst-case efficiency class. (5)

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

If the algorithm you designed is in $\Theta(n^2)$, then design a linear algorithm for this problem. Is it possible? Justify your answer with proper explanation.