No. of Printed Pages : 5

BICS-029

DIPLOMA – VIEP – COMPUTER SCIENCE AND ENGINEERING (DCSVI) / ADVANCED LEVEL CERTIFICATE COURSE IN COMPUTER SCIENCE AND ENGINEERING (ACCSVI)

Term-End Examination

00197

December, 2017

BICS-029 : ALGORITHMS AND LOGIC DESIGN

Time : 2 hours *Maximum Marks : 70*

Note : *Attempt **five** questions in all. Question number **1** is **compulsory**. Each question carries equal marks.*

1. Choose the correct answer from the given alternatives : *7×2=14*

(a) Which of the following algorithms scans the list by swapping the entries, whenever a pair of adjacent keys are out of desired order ?

(i) Insertion Sort

(ii) Quick Sort

(iii) Shell Sort

(iv) Bubble Sort

(b) The time complexity of the shortest path algorithm can be bounded by

(i) $O(n^2)$

(ii) $O(n^4)$

(iii) $O(n^3)$

(iv) $O(n)$

(c) Worst case efficiency of _____ Search is O(n).

(i) Sequential

(ii) Binary

(iii) Indexed

(iv) Hashing

(d) For a Bubble Sort algorithm, what is the time complexity of the best/worst case ? (Assume that the computation stops as soon as there are no more swaps in one pass)

(i) best : O(n), worst : $O(n^2)$

(ii) best : O(n), worst : O(n log n)

(iii) best : O(n log n), worst : O (n log n)

(iv) best : O(n log n), worst : $O(n^2)$

(e) For the Quick Sort algorithm, what is the time complexity of the best/worst case ?

(i) best : O(n), worst : $O(n^2)$

(ii) best : O(n), worst : O(n log n)

(iii) best : O(n log n), worst : O (n log n)

(iv) best : O(n log n), worst : $O(n^2)$

(f) What term is used to describe an O(n) algorithm ?

(i) Constant

(ii) Linear

(iii) Logarithmic

(iv) Non-Polynomial Deterministic

(g) The time complexity of binary search in best and worst cases for an array of size N is

(i) best : N, worst : N^2

(ii) best : 1, worst : log N

(iii) best : log N, worst : N^2

(iv) best : 1, worst : N log N

2. (a) What are the principles of recursion ? Write a recursive and iterative algorithm to find the Greatest Common Divisor (GCD) of two numbers X and Y i.e., GCD(X, Y). 7

(b) For a given problem P, two algorithms A1 and A2 have respective time complexities $T_1(n)$ and $T_2(n)$ in terms of size n, where $T_1(n) = 4n^5 + 3n$ and $T_2(n) = 2500\ n^3 + 4n$. Find the range for n and the size of an instance of the given problem, for which A1 is more efficient than A2. 7

3. (a) Write the Quick Sort algorithm. Determine the complexity of Quick Sort algorithm in best case, average case and worst case. Sort the sequence

5, 11, 21, 6, 14, 8, 12, 28, 32

in increasing order using Quick Sort. *10*

(b) Prove that worst case for Quick Sort is best case for Bubble Sort. *4*

4. (a) Explain Selection Sort algorithm with a suitable example. Calculate its complexity in best, average and worst cases. *7*

(b) Write a recursive and iterative algorithm to determine the sum of n numbers (from 1 to n). Compare the space and time complexities of the algorithm, written by you, and suggest which mechanism of algorithm writing is better, recursive or iterative ? *7*

5. (a) Write the Bucket Sorting algorithm. Sort the following list using the Bucket Sort technique : *7*

A = < 0·78, 0·17, 0·39, 0·26, 0·72, 0·94, 0·21, 0·12, 0·23, 0·66 >

(b) What do you mean by order of growth of the running time of an algorithm ? What are the asymptotic notations ? Explain Big-Oh and Big-Omega notations. *7*

6. (a) Write the recursive expression to generate a Fibonacci series. Use the expression to generate the first five terms of the series. Design a flow chart to implement a Fibonacci series. 7

(b) Explain the search mechanism of Fibonacci search with suitable example. 7

7. (a) Compute the space and time complexities of any ***two*** of the following : 7

(i) Sequential Search

(ii) Binary Search

(iii) Fibonacci Search

(b) Write the algorithm for Insertion Sort and apply this algorithm to sort the following data : 7

10, 5, 9, 13, 8, 12

8. Write short notes on the following : $4 \times 3\frac{1}{2} = 14$

(a) Asymptotic Notations

(b) Program Development Cycle

(c) Characteristics of a Good Algorithm

(d) Process of Validation of Algorithm