**2017/ODD/08/24/MCS–702 (C)/**
**MS–102 (C)/129**

**PG Odd Semester (CBCS) Exam., December—2017**

COMPUTER SCIENCE

**( 7th Semester ) / ( 1st Semester )**

Course No. : MCS–702 (C)/MS–102 (C)

**( Design and Analysis of Algorithms )**

*Full Marks* : 75
*Pass Marks* : 30

*Time* : 3 hours

*The figures in the margin indicate full marks for the questions*

Answer **five** questions, selecting **one** from each Unit

UNIT—I

**1.** *(a)* What are $O$, and notations? Why is called a tight bound? 5

*(b)* Explain why the statement—"The running time of algorithm $A$ is at least $O(n^2)$" is meaningless. 2





*(c)* Are the following correct? Justify your answers : 2×4=8

*(i)* $2^{2n}$ $O(2^n)$

*(ii)* $10n^3$ $n\log n$ $(n^3)$

*(iii)* $33n^3$ $4n^2$ $(n^2)$

*(iv)* $n^{2^n}$ 6 $2^n$ $(n^{2^n})$

**2.** *(a)* Show that if $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$, then $f_1(n)$ $f_2(n)$ is $O(\max(g_1(n), g_2(n)))$. 3

*(b)* Give asymptotic bounds for the following recurrences : 3×4=12

*(i)* $T(n)$ $2T(\sqrt{n})$ $\lg n$

*(ii)* $T(n)$ $T(n$ 1) $n$

*(iii)* $T(n)$ $2T(n/4)$ $n^2$

*(iv)* $T(n)$ $2T(n/2)$ $n\lg n$

UNIT—II

**3.** *(a)* What is the number of element comparisons if a naive algorithm for finding maximum and minimum of $n$ elements is used? 3

*(b)* Show that the number of element comparisons for an efficient maxmin algorithm is $3n/2$ 2. 6

*(c)* Show that the time taken by merge sort on an array of size $n$ is $O(n\log n)$. 6

**4.** *(a)* Show that the worst case behaviour of quick sort occurs when the array is completely sorted. Also show that quick sort takes time $O(n \log n)$ when the middle element is chosen as the pivot ($n$ is the number of elements in the array). 6

*(b)* What is a heap? What are the maximum and minimum number of elements in a heap of height $h$? 4

*(c)* Show that an $n$-element heap has height $\lg n$ . 5

UNIT—III

**5.** *(a)* How are graphs represented computationally? Elucidate the advantages and disadvantages of both the approaches. 2+4=6

*(b)* Explain how BFS can record distances from a start node to every other node in a graph. When does BFS compute shortest paths? Show that the running time for recording distances by BFS using adjacency list representation is $O(n \quad m)$, where $n$ is the number of vertices and $m$ is the number of edges in the graph. 4+2+3=9





**6.** *(a)* What is a directed acyclic graph (DAG)? Show that every DAG can be topologically sorted. 2+5=7

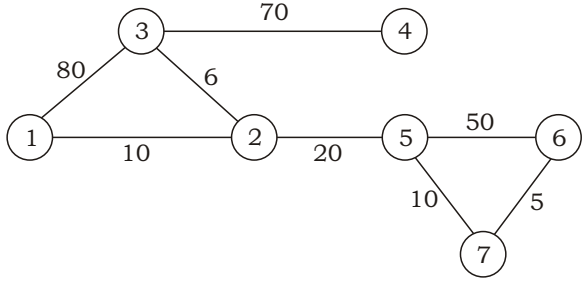
*(b)* Show how the longest path in a DAG can be found. Also give an efficient linear time algorithm for finding it. 8

UNIT—IV

**7.** *(a)* Explain greedy algorithms. How does it drastically reduce space to search for solutions? 3+2=5

*(b)* How can you classify Dijkstra algorithm as a greedy algorithm? Show that the relaxation operation maintains the invariant that $d[\upsilon] \quad (S, V)$ for all $\upsilon \quad V$, where $V$ is the set of vertices. 2+4=6

*(c)* Find the shortest path from node 1 to every other vertex in the figure given below using Dijkstra algorithm : 4



**8.** *(a)* What is optimal substructure property? 3

*(b)* What is memorization? How is it an improvement over recursion? 3

*(c)* What is dynamic programming? Show the working of dynamic programming on the longest common subsequence problem. 2+7=9

UNIT—V

**9.** *(a)* Explain what is reduction. 3

*(b)* What are SAT and 3-SAT problems? 3

*(c)* Argue that 2-SAT is in *P*. 3

*(d)* Show how independent set and vertex cover problems are mutually reducible. 6

**10.** *(a)* Elucidate *NP*-complete and *NP*-hard problems. 5

*(b)* Show how SAT can be reduced to 3-SAT. 4

*(c)* Show how 3-SAT problem can be reduced to independent set problem. 6

★ ★ ★