Exam.Code:0999
Sub. Code: 34961

2124

M.E. (Computer Science and Engineering)
First Semester
CS-8101: Advance Algorithms
(Common with CSN 8101)
(For UIET)

Time allowed: 3 Hours **Max. Marks: 50**

NOTE: Attempt <u>five</u> questions in all, including Question No. 1 (Section-A) which is compulsory and selecting two questions from each Section B-C.

x-x-x

Section-A

Q1 (a) Why do we should not look only at worst case analysis? What are different methods for analyzing recurrence algorithms? (2)

(b) What are different methods for analyzing recurrence algorithms? (2)

(c) Given the following functions $f_1 = \log(n!)$, $f_2 = (logn)!$, $f_3 = n^{\log(n)}$, $f_4 = \log(\log(n))$. Arrange these functions in increasing order of their growth. (2)

(d) What is the difference between NP hard and NP complete problems? (2)

(e) What is the time complexity of string matching with finite automata and compare it with the naïve string matching algorithm. (2)

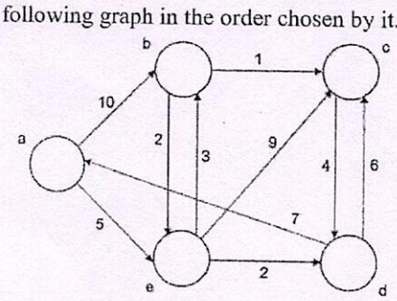
Section-B

Q2 (a) Discuss the correctness of following equalities (5)
(i) $n! = O(n^n)$
(ii) $2^n + n \log n + 5 = \theta(n^3)$

(b) Solve the recurrence relation $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n^2)$ is (5)

Q3 (a) Explain the strassen's matrix multiplication algorithm in detail. Compare its time complexity with other matrix multiplication algorithms. (5)

(b) Explain Prim's algorithm and describe the data structures used in its implementation. List the costs of the edges of following graph in the order chosen by it. (5)



Q4 (a) Discuss the significance of Lower Bound theory for algorithm designers. Draw the comparison tree for binary search. Prove that minimum comparisons required for any comparison-base algorithm on n-elements is more than log(n+1). (5)

(b) Discuss the greedy method to solve the knapsack problem for instance (5)
n=7 m=15, (P1, P2, … P7)= (10, 5, 15, 7, 6, 18, 3) and
(W1, W2 … W7) = (2, 3, 5, 7, 1, 4, 1)
Is this the optimal method?

Section-C

Q5 (a) State the principal of optimality. How dynamic programming is different from the greedy programming? Write an algorithm to solve Traveling Salesman problem using Dynamic Programming. (5)

(b) Define Multistage Graph and discuss an algorithm to find the cost path from source to sink using dynamic programming (5)

Q 6(a) Using the text T= "ABABABCABABABAC" and the pattern P="ABABAC", demonstrate the execution of the KMP algorithm step-by-step. Show how the partial match table is used to skip unnecessary comparisons. (5)

(b) Design a PRAM algorithm to color the graph $G=(V,E)$ with n vertices and m edges using the minimum number of colors such that no two adjacent vertices share the same color. (5)

Q 7 a) What are the main steps for designing an approximation algorithm? Which one is better: n-approximation, log n-approximation, $\sqrt{}$ n-approximation, constant factor approximation (5)

(b) What are NP-Scheduling problems? Explain and compare them with NP-hard and NP-complete problems. (5)

x-x-x