SPRING END SEMESTER EXAMINATION-2018

4th Semester B.Tech & B.Tech Dual Degree

DESIGN & ANALYSIS OF ALGORITHMS

CS-2008

[For 2017(L.E.), 2016 & Previous Admitted Batches]

Time: 3 Hours | Full Marks: 60

Answer any SIX questions including question No.1 which is compulsory.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

1. (a) Write the time complexity for finding the 3rd maximum in an (unordered) array of n distinct elements. [2 × 10]

   (b) Write the time complexity for addition of two matrices of size $m \times n$ each.

   (c) Write the asymptotic tight bound of $n \log(n!)$.

   (d) $A[1..n]$ is a list of n distinct numbers. Given that x belongs A, $\sum_{i=1}^{n}\left|A[i]-x\right|$ is minimum if x is: (a) mean (b) median (c) some element of A between (and including) mean and median (d) not necessarily any of these.

   (e) Define binary max heap. Name an operation which is efficient on a binary max heap.

   (f) Minimum number of comparison required to merge two sorted files of size m and n is MIN(m, n). How many minimum comparisons are required to merge 3 sorted files of size 10, 20, and 30?

   (g) Write the worst-case and the average-case time complexities of heap sort.

(h) Write the space complexities for adjacency-list representation and adjacency-matrix representation of an arbitrary tree having n nodes.

(i) Which one among the following problems is easiest and which one hardest (measured in terms of worst-case time complexity of the best-known algorithm)?

P1: Finding the maximum in an unordered list.

P2: Finding the minimum in an unordered list.

P3: Searching a key in an ordered list.

P4: Searching a key in an unordered list.

P5: Finding a pair of elements in an unordered list such that their difference is minimum.

P6: Partitioning an unordered list into sub-lists such that:

- they are of equal size or differs in size by at most one;
- each element in the first sub-list is less than eachelement in the second sub-list.

(Assume that every time the list is implemented by a 1-D array and contains n distinct elements.)

(j) Which of the following problems is/are *NP*-Complete?

P1: Matrix Chain Multiplication Problem

P2: Minimum Spanning Tree Problem

P3: Circuit Satisfiability Problem

P4: Binary Knapsack Problem

P5: Traveling Salesman Problem

P6: Median Finding Problem

2. (a) Deduce the asymptotic tight bound, if possible, for [4]

$$T(n) = \begin{cases} \Theta(1) & \text{if } n < 40, \\ 4T(\lfloor n/5 \rfloor) + n & \text{otherwise.} \end{cases}$$

If asymptotic tight bound is not possible, then justify the reason and deduce the upper asymptotic bound.

(b) State and explain Master's Method and use this method to give tight asymptotic bounds for the recurrence $T(n) = 4T(n/2) + n^3$. [4]

3. (a) Derive the average-case time complexity for linear search in an unordered list. [4]

(b) Write the algorithm for insertion sort and use step count method to analyze the time complexities of Insertion Sort. [4]

4. Given a set of n intervals, $R = \{I_1, I_2, \ldots, I_n\}$, where each $I_j = [a_j, b_j]$ is an interval from the real line having length $b_j - a_j$. The objective is to find the largest set of intervals $S \subseteq R$ such that no two intervals in S overlap each other and the sum of lengths of the intervals in S is maximized. Suggest an efficient algorithm for this and derive its time complexity. [5+3]

5. $3n$ distinct elements are given in a one-dimensional array A. Suggest an efficient algorithm to partition A into B, C, D, such that i) each of B, C, D contains n elements, ii) each element of B is less than each element of C, and iii) each element of C is less than each element of D. Derive its time complexity. [5+3]

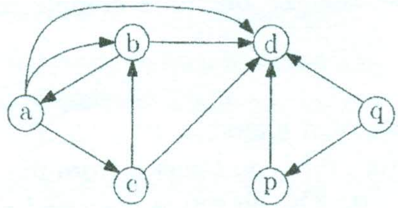
6. (a) Find the optimal parenthesization of a Matrix-Chain product for minimizing total number of scalar multiplications, whose sequence of dimensions is $< 5\times10, 10\times8, 8\times7, 7\times6, 6\times5 >$. [4]

(b) Use Dynamic Programming Algorithm to find the Longest Common Subsequence of the following two sequences: [4]

$X = < ababaabaa >$

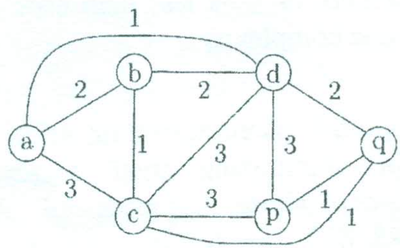
$Y = < aababaab >$

7. Construct the adjacency list of the following directed graph and demonstrate the DFS (depth-first search) algorithm on it. Write the initialization and explain how the relevant parameters and data structures are updated during the execution. In the final step, you should write the DFS tree/trees, and also the forward edges, cross edges, and back edges, if any. Use node '*a*' as source node while answering the question. [5+3]



8. (a) Construct the adjacency list of the following weighted graph and demonstrate on it the Kruskal's algorithm for minimum spanning tree. Explain how the relevant parameters and data structures are updated during the execution. In the final step, you should write the MST and its cost. Is this MST unique for this graph? [5]

Use node '*a*' as source node while answering the question.



(b) Define the classes *P* and *NP*. Mention one problem that belongs to *NP* but not to *P*. [3]

*****