Total No. of Questions : 10] [Total No. of Printed Pages : 7

Roll No. .....

CS/IT–404

B. E. (Fourth Semester)
EXAMINATION, June, 2012

(Grading/Non-Grading)

(Common for CS & IT Engg. Branch)

ANALYSIS AND DESIGN OF ALGORITHMS

Time : Three Hours

Maximum Marks : $\begin{cases} GS : 70 \\ NGS : 100 \end{cases}$

Note : Attempt any *five* questions. All questions carry equal marks.

1. (i) How many average comparisons are required by the following sorting algorithms to sort the only k highest elements, out of n elements ?
Bubble sort, Quick sort, Insertion sort, Merge sort, Heap sort.

(ii) Solve the following recurrence relation :

(a) $T(n) = T(\sqrt{n}) + n$

(b) if $n >= 2$ otherwise $T(n) = 1$.

Or

2. (i) Write an algorithm for calculation the Fn Fibonacci number in $\Theta(\log n)$ time.

(ii) Show that the second smallest of n elements can be found with $n + [\log n] - 2$ comparisons in the worst case.

3. (i) At lunchtime it is crucial for people to get to the food as quickly as possible. Consider the following model. The building is represented by a graph $G = (V, E)$, where each room, landing, or other location is represented by a vertex and each corridor or stairway is represented by an edge. Each corridor has an associated capacity c, meaning that at most c people can pass through the corridor at once. Traversing a corridor from one end to the other takes one time step. (Traversing a room takes zero time) :

(a) Suppose all people are initially in a single room s, and that the building has a single exit t. Show how to find a fastest way to get everyone out of the building.

(b) Show that the same technique can be used when people are initially in multiple locations and there are multiple exits.

(c) Suppose transit times are not integral. Is there still a way to solve the problem ?

The above algorithm is polynomial in the number of people. Can you improve it to be polynomial (in the graph size) regardless of the number of people ?

(ii) A table composed of N × M cells, each having a certain quantity of apples, is given. You start from the upper-left corner. At each step you can go down or right one cell. Design an algorithm to find the maximum number of apples you can collect, if you are moving from uppe-left corner to bottom-right corner.

Or

4. (i) Given a set of n men and n women. Assume each member of these sets has a preference list of partners that may include indifferences. Each man and woman will have a distinct first choice, but in any of their remaining choices they may be indifferent between two or more of the selections. For instance if $n = 4$, man m_1 may prefer woman w_1 as his first choice, but may consider his second choice to be tied between w_3 and w_4 with w_2 his distinct least favorite option. A strong instability in a perfect matching S consists of a man m and a woman w, such that each of m and w prefers the other to their partner in S. Does there always exist a perfect matching with no strong instability in every instance of the stable matching problem with indifferences ? A weak instability in a perfect matching S consists of a man m and woman w with partners w' and m′ in S respectively such that (a) m prefers w to w' and w either prefers m to m' or is indifferent between these two, or (b) w prefers m to m', and m either prefers w to w' or is indifferent

between the two choices. Design and analyze an algorithm to find a perfect matching with no weak instabilities ?

(ii) Prove that the fractional Knapsack problem has the greedy-choice property.

5. (i) A company is planning a party for its employees. The organizers of the party want it to be a fun party, and so have assigned a 'fun' rating to every employee. The employees are organized into a strict hierarchy, i. e., a tree rooted at the President. There is one restriction, though, on the guest list to the party : both an employee and their immediate supervisor (parent in the tree) cannot both attend the party (because that would be no fun at all). Give best an algorithm that makes a guest list for the party that maximizes the sum of the 'fun' ratings of the guests.

(ii) Let a truck that can carry no more than t pounds and n different boxes to ship, the ith box of which weight x_i pounds. Design and analyze an algorithm to fill the truck with as heavy a load as possible without exceeding the given weight limit. If the complexity of yours algorithm in oder of exponential then design an approximation algorithm for same.

Or

6. (i) Write an algorithm for the longest common subsequence (LCS). Also determine the LCS of < A, B, C, B, D, A, B > and < B, D, C, A, B, A >.

(ii) Write an algorithm for matrix-chain multiplications. Also find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is <5, 10, 3, 12, 5, 50, 6>.

7. (i) You are given an unlimited number of each of n different types of envelopes. The dimensions of envelope type i are $x_i \times y_i$. In nesting envelopes inside one another, you can place envelope A inside envelope B if and only if the dimensions A are strictly smaller than the dimensions of B. Design and analyze an algorithm to determine the largest number of envelopes that can be nested inside one another.

(ii) You are given a sequence of n integers that contains $\log n$ different integers. Design an algorithm to sort this sequence using $O(n \log \log n)$ element comparions.

Or

8. (i) Consider the following puzzle played on an $n \times n$ square grid, where each square is labelled with a positive integer. A token is placed on one of the squares. At each turn, you may move the token left, right, up or down; the distance you move the token must be equal to the number on the current square. For example, if the token is on a square labelled "3", you are allowed move the token three squares down, three square left, three squares up, or three squares

right. You are never allowed to move the token off the board :

(a) Describe and analyze an algorithm to determine, given an $n \times n$ array of labels and two squares s and t, whether there is a sequence of legal moves that takes the token from s to t.

(b) Suppose you are only given the $n \times n$ array of labels. Describe how to preprocess these values, so that afterwards, given any *two* squares s and t, you can determine in O(1) time whether there is a sequence of legal moves from s to t.

(ii) In order to facilitate recompiling programs from multiple sources files when only a small number of files have been updated, there is a UNIX utility called 'make' that only recompiles those files that were changed after the most recent compilation, and any intermediate files in the compilation that depend on those that were changed. A Makefile is typically composed of a list of source files that must be compiled. Each of these source files is dependent on some of the other files which are listed. Thus a source file must be recompiled if a file on which it depends is changed. Assuming you have a list of which files have been recently changed, as well as a list for each source file of the files on which it depends, design an algorithm to recompile only those necessary.

9. (i) Show that the subset-sum problem is solvable in polynomial time if the target value t is expressed in unary.

(ii) Given an array of numbers, except for one number all the others occur twice. Give a tree/graph based algorithm to find that number which occurs only once in the array.

Or

10. (i) Show that Hamiltonian cycle problem is NP-complete.

(ii) For a given binary tree, design an algorithm to verify it is a binary search tree or not. Also find the complexity of your algorithms.