Q. No. IT - 504 / 101

B. Tech./Odd
2016-17/Reg

2016-17

DESIGN AND ANALYSIS OF ALGORITHMS

IT - 504

Full Marks : 70 Time : Three Hours

The figures in the margin indicate full marks.

Answer question number 1 and any *five* from the rest.

Answer them in order.

1. (a) Where in a *max-heap* might the smallest element reside, assuming that all elements are distinct? Justify. 2

(b) Why the runtime of *max-heapify* is bounded by

$T(n) \le T(2n/3) + \theta(1)$

Solve the recurrence using master method. 1+2

(c) Why do we bother setting the key of the inserted node to — ∝ in the *max-heap-insert* when the next thing we do is to increase its key to the designed value? 2

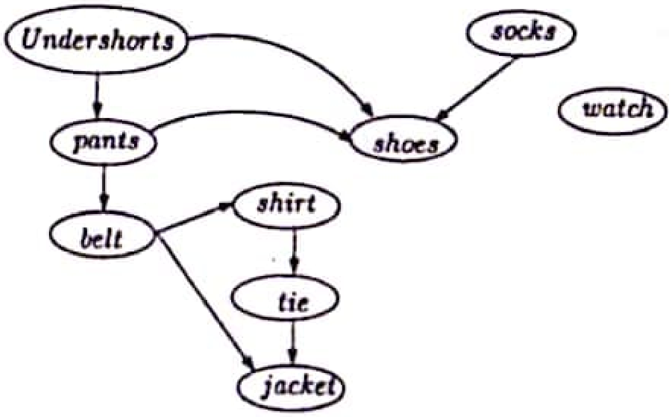
(d) Give an $O(n \lg n)$ time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all. Write a pseudocode for that. 5

(e) Prove that the *Build-max-heap* algorithm is correct by the *loop-invariant* technique. 4

P.T.O.

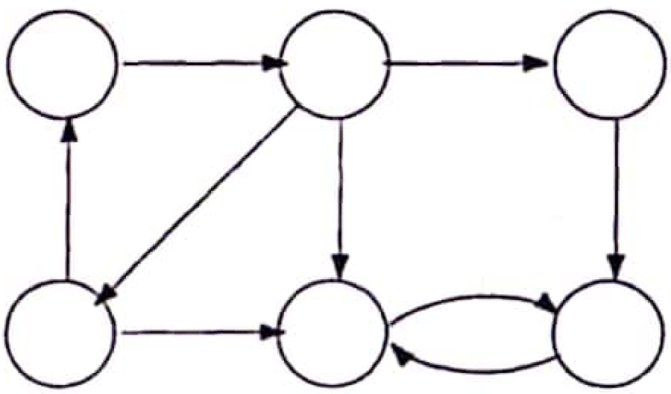
(f) Prove that the running time of BFS procedure is $O(V+E)$ by the Amortized analysis technique. 4

(g) Say Prof. Robert wants to decide the order in which he should be dressed when the following natural constraints are imposed.



Give an algorithm that will give a proper ordering for his dressing sequence. Analyse the time complexity of your algorithm. Illustrate the procedure. 5

2. Consider the following graph. You can give the name of the nodes according to your own will.



Write an algorithm to find the strongly connected components of the graph. Illustrate the procedure with the above graph. Write the time complexity of your algorithm. 3+3+3

3. (a) Which data structure is used to implement Kruskal's algorithm? Why do we use that? 1+1

   (b) What are the two important heuristics that are used to reduce the time complexity of different operations of that data structure ? Discuss with examples. Write the pseudocode of different operations that are supported in that data structure. 2+2+3

4. (a) What are the different *greedy templates* for Interval scheduling problem ? 2

   (b) Give counter examples for at least three of them, so that it can be shown that they are not optimal. 3

   (c) Give a pseudocode for the above interval scheduling problem so that it can be solved optimally. Prove by the exchange argument that your algorithm is optimal. 4

5. (a) Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted order. Give an $O(\lg n)$-time algorithm to find the median of all $2n$ elements in arrays X and Y. Write the pseudo-code. 5

(b) From which problem does the following recurrence come?

$$T(n) \le \begin{cases} O(1) & \text{if } n < 140 \\ T\left(\left\lceil \frac{n}{5} \right\rceil\right) + T\left(\frac{7n}{10} + 6\right) + O(n) & \text{if } n \ge 140 \end{cases}$$

Solve the recurrence by substitution method. 4

6. (a) Prove that the direct recursive implementation of Matrix-chain-multiplication without memorization yields the running time $T(n) = \Omega(2^n)$. 4

(b) Draw the recursion tree of solving the problem with chain length at least 5. Highlight the overlapping sub-problem property in the recursion tree. 3

(c) Describe the main observation by which Karatsuba got his improved multiplication algorithm. 2

7. (a) When does the Frod-Fulkerson method performs badly? Illustrate with an example. 2

(b) How does the algorithm be improved? What is the name of that algorithm? 2

(c) With an example show how does the Edmond-Carp algorithm behaves on an input graph. Analyse the time-complexity of Edmond-Carp algorithm. 3+2

2015-16

DESIGN AND ANALYSIS OF ALGORITHMS

IT 504

Full Marks : 70 *Time* : 3 hours

Figures in the margin indicate full marks of each question.

Answer all questions and in order.

1. (*a*) Say there are several cars moving in the street. How can you minimize the number of collisions that may occur between two cars. Give an efficient algorithm for this problem. First, characterize the problem and then write the algorithm. Analyze the time complexity of your algorithm (where ever necessary provide the proof). 3 + 4 + 4

   (*b*) Prove that

$$\begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n$$

   where F_n is the n^{th} Fibonacci Number. 3

2. (*a*) Let us consider the following recurrence :

$$T(n) \le \begin{cases} 0(1) & \text{if } n < 140 \\ T(\lceil n/5 \rceil) + T(7n/10+6) + 0(n) & \text{if } n \le 140 \end{cases}$$

   from which algorithm, the recurrence is generated ? Define the problem and write the algorithm. Can you solve this recurrence by using Master method ? Justify your logic. Solve the recurrence. 1 + 1 + 3 + 3 (*b*) Given a set of n numbers, we wish to find the i largest in sorted order using a comparison based algorithm. Find the algorithm that implements each of the following methods with the best asymptotic worst case running time, and analyze the running time of the algorithm in terms of n and i.

(*i*) Sort the number, and list the i largest.

(*ii*) Use an order statistic algorithm to find the i^{th} largest number, partition around that number and sort the i largest numbers. 6

3. (*a*) Solve the following recurrences using Akra-Bazzi method.

(*i*) $T(x) = T(x/2) + \Theta(\lg x)$. 3

(*ii*) $T(x) = 1/2T(x/2) + \Theta(1/x)$. 3

(*b*) For counting the number of parenthesization in matrix chain multiplication problem. Consider the following recurrence.

$$p(n) = \begin{cases} 1 & \text{if } n = 1 \\ \sum_{k=1}^{n-1} P(k)p(n-k) & \text{if } n \geq 2 \end{cases}$$

use the substitution method to show that the solution to the recurrence is $\Omega(2^n)$. 3

(*c*) Find an optimal parenthesization of a matrix chain product whose sequence of dimension is <2,4,2,6,3>. You don't have to write the algorithm. 5

4. (*a*) By loop invariant technique prove that the following algorithms are correct :

(*i*) The partitioning algorithm which is at the heart of quick sort,

(*ii*) the algorithm for Horner's rule. 4 + 4

(*b*) Prove that the greedy-set-cover is a polynomial time $\rho(n)$– approximation algorithm,

where $\rho(n) = H(\max\{|S| : S \in \mathcal{F}\})$. 6

5. (*a*) How cut-and-paste technique can be utilized in algorithmic analysis : Give one example where we can utilize this concept. Prove the same. 3

(*b*) Why do we use Amortized analysis technique ? Analyze the following algorithms using the aggregate analysis technique :

(*i*) Breadth first search.

(*ii*) Prefix function computation in KMP string matching.

Write the algorithm and then analyze. 5 + 6

———

2014-15

DESIGN AND ANALYSIS OF ALGORITHMS

IT - 504

Time - Three Hours Full Marks - 70

Answer ***all*** questions and in order.

Figures in the margin indicate full marks of each question.

1. (a) Prove that the running time of insertion-sort, in the worst case is $O(n^2)$. 5

(b) Prove that the insertion-sort algorithm is correct with the loop-invariant technique. 5

(c) Give a linear time algorithm for computing the value of a polynomial, defined below, at some point x.

$$P(x) = \sum_{k=0}^{n} a_k x^k .$$

Prove that your algorithm is correct with the loop-invariant technique. 5

2. Let $A[1...n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A.

(a) List the five inversions of the arrays

$< 2, 3, 8, 6, 1 >$. 2

(b) Give an algorithm that determines the number of inversions in any permutation on n elements in $\Theta(n lg n)$ time by modifying the merge sort. Show the time-complexity analysis. 6

(c) Suppose that the elements of A form a uniform random permutation of $< 1, 2, \ldots, n >$. By using indicator random variable compute the expected number of inversions in this case. 4

(d) Given a sample space S and an event A in the sample space S, let $X_A = I\{A\}$. Then, prove that $E[X_A] = Pr\{A\}$. 3

3. (a) Sort the following group of functions in increasing order of asymptotic (big-O) complexity. Justify your answer.

$$f_1(n) = 2^{2^{10000}}$$
$$f_2(n) = 2^{10000n}$$
$$f_3(n) = \binom{n}{2}$$
$$f_4(n) = n\sqrt{n}$$

4

(b) How to guess a solution approximately correctly using recursion tree ? Illustrate with an example where you can apply the concept of infinite series. 5

(c) How can the divide and conquer approach be used to solve the maximum-subarray-sum problem ? Is it the optimum solution in terms of the upper bound achieved ? Give a linear time algorithm for that. 3+3=6

4. (a) What is the utility of Kruskal's algorithm ? Write the algorithm and analyze the running time of the Kruskal's algorithm. 1+2+3=6

(b) Say there is a problem of scheduling a single lecture room for several competing lecture tasks. For each task, there is a start time and a finish time. The goal is to schedule the tasks, so that maximum number of tasks can be scheduled. Characterize the problem first, and then provide a greedy solution for the problem. Prove that your greedy choice leading to an optimal solution by using the cut and paste technique. Write the algorithm and analyze its time complexity. 2+3+2+2=9

5. (a) How does the Homer's rule come in solving the Rabin-Karp string matching algorithm? Illustrate the Rabin-Karp algorithm with an example. 2+2=4

(b) Write the Knuth-Morris-Pratt (KMP) string matching algorithm. Analyze its time complexity by using the amortized analysis technique. 2+4=6