HERITAGE INSTITUTE OF TECHNOLOGY

Class Test I Examination 2017 Session: 2016 - 2017

Discipline: Computer Science & Engineering

Paper Code: CSEN 2201 Paper Name: Design & Analysis of Algorithms Date: 8th March 2017

Time – 70 min Total Marks: 30

Total marks is 36 but the maximum you can score is only 30

Group – A (Linearly Easy - 14 marks – approx 25 min)

1. Choose the best possible alternative for the following questions: (4x1 = 4)

(i) Worst case time complexity for inserting an element in a sorted array so that it stays sorted is

(a) O(1) (b) O(n) (c) $O(n^2)$ (d) None of these

(ii) An algorithm is made up of 2 modules M1 & M2. If order of M1 is f(n), M2 is g(n), then the order of the algorithm is

(a) max(f(n), g(n)) (b) min(f(n), g(n)) (c) f(n) + g(n) (d) f(n) * g(n)

(iii) The best-case time complexity for binary search from an array of n sorted elements is

(a) O(n) (b) O(log n) (c) O(n / log n) (d) O(1)

(iv) The lower bound for comparison sort is

(a) $O(n \log_{10} n)$ (b) $O(n \log_2 n)$ (c) $O(n \log_e n)$ (d) all of the above.

2. Given an array of n integers. Give an algorithm to print the leaders in the array in O (n) time. A leader is an element which is larger than all the elements in the array to its right. In array {98, 23, 54, 12, 20, 7, 27 }, leader elements are 98, 54, 27. Note that the last element is always a leader since there is no element to its right. (3)

ANS: Just write a single pass (for-loop) from the end. Keep a variable called 'leader'. Load the last element in it and print it as leader. Then keep comparing every element with leader, if it is greater then make it the new leader and print it. If someone prints for greater than or equal to condition, may be given 2.5 marks.

3. Write down the basic principles of any Divide-and-Conquer method. Show that if we do an average-case analysis, the time T(n) required for sorting n numbers using Quicksort can be expressed as the following recurrence relation

$$T(n) = (2/n)\Big(\sum_{1 \le k \le n-1} T(k) \Big) + \theta(n)$$

where $\theta(n)$ has its usual meaning w. r. t. asymptotic complexity. (5)

HERITAGE INSTITUTE OF TECHNOLOGY

Class Test I Examination 2017 Session: 2016 - 2017

Discipline: Computer Science & Engineering

Paper Code: CSEN 2201 Paper Name: Design & Analysis of Algorithms Date: 8th March 2017

Time – 70 min Total Marks: 30

Total marks is 36 but the maximum you can score is only 30

Ans: The chance of the pivot having a rank r, 1<= r <= n, is equally likely assuming the numbers coming from a uniform distribution and hence the probability for each case is (1/n).

$$T(n) = (1/n) \sum_{1 \le k \le n-1} (T(k-1) + T(n-k)) + \theta(n)$$

And then some algebraic manipulation.

4. Matrices A, B and C, of dimensions 10 x 25, 25 x 6 and 6 x 15 respectively, are provided. You are told to ensure that when computing the matrix product A.B.C the number of scalar multiplications will be a minimum. How would you parenthesize the product, and what would be the minimum number of scalar multiplications you would need to perform? (2)

Ans. (AxB)xC means (10x 25x 6) + 10 x 6 x 15 = 1500 + 900 = 2400 (Minimum)

(A x (B x C)) means 25 x 6 x 15 + 10 x 25 x 15 = 2250 + 3750

Group – B (Quadratically Mediocre - 12 marks – approx 25 min)

5. Solve **either a) OR b)** (4)

(a) Consider the definition

```
h(0) = 1
h(n) = h(n-1) + h(floor(n/2)) + 1
```

For example, h(2) = h(1) + h(1) + 1 = 3, h(3) = h(2) + h(1) + 1 = 5. If you write a recursive implementation of the above definition, will it be efficient? If yes, can you find its time complexity? If not, how can the algorithm be improved? Justify your answer.

HERITAGE INSTITUTE OF TECHNOLOGY

Class Test I Examination 2017 Session: 2016 - 2017

Discipline: Computer Science & Engineering

Paper Code: CSEN 2201 Paper Name: Design & Analysis of Algorithms Date: 8th March 2017

Time – 70 min **Total Marks: 30**

Total marks is 36 but the maximum you can score is only 30

Ans. This is not an efficient algorithm because it recomputes expressions. For example, computing h(4) requires computing h(3) and h(2). But computing h(3) requires computing h(2) again. The algorithm can be improved by storing the values h(i) in an array. For i = 1,...,n, compute h[i] according to the equation, looking in the array to find h(i-1) and h(floor(i/2)). Finally, get the value h[n] from the array.

(b) You are provided as input an integer $n > 1$ and a permutation $p_1p_2\ldots p_n$ of the integers $12\ldots n$. Describe **briefly** a method that will find the longest increasing subsequence in the given permutation in time $O(n^2)$ and space $O(n^2)$. For example, if n is 9 and the permutation is (6 3 8 7 1 4 2 5 9), the subsequence (3 4 5 9) is strictly increasing and has length 4. There is no strictly increasing subsequence of length 5, but there are many other strictly increasing subsequences of length 4, and any of them would be acceptable as the solution.

Ans. Sort the sequence in ascending order in nlog n time. So it is then an **increasing** (non-decreasing) sequence. Now find the LCS between this sorted sequence and the original sequence using the standard algorithm to find LCS, to get your answer in $O(n^2)$ time.

(No more explanation needed to get full marks)

6. Solve **either a) or b)** (4)

a) Solve the following recurrence relation-

$$T(n) = 2T(\sqrt{n}) + \log n$$

Renaming m = lg n yields,

$T(2^m) = 2T(2^{m/2}) + m$,We can now rename,

$S(m) = T(2^m)$ to produce the new recurrence,

$$S(m) = 2S(m/2) + m$$

Indeed, the above recurrence has the solution: $S(m) = O(m \lg m)$

Changing back from S(m) to T(n), we got T(n) = O(log n log (log n))

b) Use a recursion tree to determine a good asymptotic upper bound of the following recurrence:

HERITAGE INSTITUTE OF TECHNOLOGY

Class Test I Examination 2017 Session: 2016 - 2017

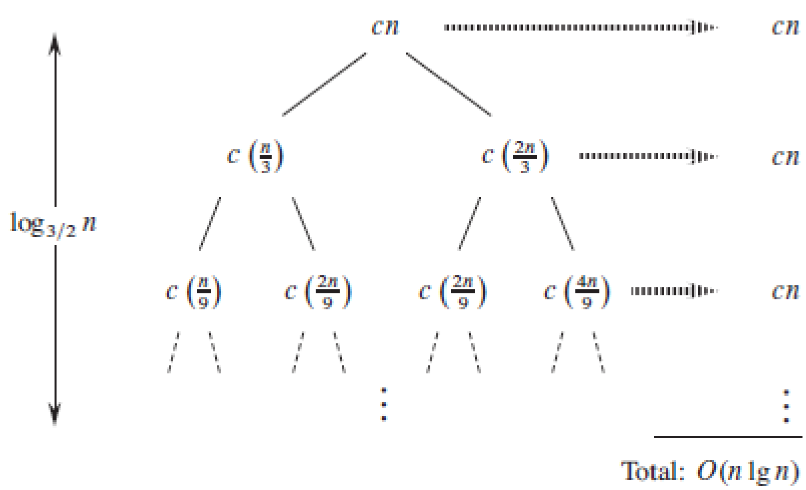
Discipline: Computer Science & Engineering

Paper Code: CSEN 2201 Paper Name: Design & Analysis of Algorithms Date: 8th March 2017

Time – 70 min **Total Marks: 30**

Total marks is 36 but the maximum you can score is only 30

T(n) = T(n/3) + T(2n/3) + cn



7. Solve **either a) or b)**
(4)

a) You are provided as input a positive integer $n > 1$ and an unsorted sequence S of n integers. You would like to find *both* the second order statistic and the nth order statistic of S. Show that they can be found out in ($3\lfloor n/2 \rfloor + \lceil \lg n \rceil - 1$) number of pair-wise comparisons.

Ans. The 1st order statistic (means minimum element) can be found out in (n – 1) comparisons. Now if you do this as a tournament arranged as a binary tree where as many teams as possible play in the 1st level and then in the 2nd level etc. then such a tree will have a height of $\lceil \lg n \rceil$. In other words the champion (the smallest element) had to play $\lceil \lg n \rceil$ games in order be the champion. If we can track which numbers were the direct losers to the champion, then the smallest of them is the 2nd order statistic. To find the minimum of $\lceil \lg n \rceil$ numbers it will take

$\lceil \lg n \rceil$ -1 more comparisons.

Now you also have to find out the nth order statistic (maximum element). But if you take n -1 comparisons more, then you exceed the given bound. So a small trick is required. Look at the comparisons that were done at the 1st level in the earlier tournament. How many such comparisons were there? $\lfloor n/2 \rfloor$ (to take care of odd n, otherwise n/2 could suffice). Note that the smaller number from each of those comparisons is useless for finding out the nth order statistic. So I can now find out the maximum element from the rest of the n - $\lfloor n/2 \rfloor$ elements, which will take exactly n - $\lfloor n/2 \rfloor$ - 1 comparisons (N. B. we do not care in which order they are compared). Hence total number of comparisons = (n -1) + $\lceil \mathbf{lgn} \rceil$ -1 + n - $\lfloor n/2 \rfloor$ - 1 =

(2n -2 - $\lfloor n/2 \rfloor$) + $\lceil \mathbf{lgn} \rceil$ -1 <= (3 $\lfloor n/2 \rfloor$ + $\lceil \mathbf{lgn} \rceil$ - 1) .

(Actually the bound is tight for n = odd, for n = even, it is loose by 2)

N. B. Students will do 2 types of mistakes. If they 1st find out the minimum and maximum using a method that does not involve tree, then they cannot solve the 2nd order statistic efficiently. On the other hand those who will use the tree for finding the minimum so that they can find the 2nd order statistic smartly may not note that for finding the maximum they do not need n – 1 comparisons again. They can utilize the 1st level comparisons smartly.

In either case they can be given 2 out of 4.

b) Explain with example why in 0-1 Knapsack it is essential to use w as a variable along with the increasing set of items?

Ans. The capacity of the Knapsack (although by problem statement bounded by W) should be considered as a variable because in that way, it is possible

HERITAGE INSTITUTE OF TECHNOLOGY

Class Test I Examination 2017 Session: 2016 - 2017

Discipline: Computer Science & Engineering

Paper Code: CSEN 2201 Paper Name: Design & Analysis of Algorithms Date: 8th March 2017

Time – 70 min Total Marks: 30

Total marks is 36 but the maximum you can score is only 30

to find optimal solutions for bags of different capacities within the range (ranging from 1 to W, i.e., the capacity of the knapsack). It starts by finding smallest possible capacity and incrementally the value of w is increased up to W. At a later stage, when we try to decide whether to take an item or not we calculate the remaining capacity of the knapsack and access the optimal solution for the remaining capacity (w') of the knapsack, which is nothing but the optimal solution for a knapsack of size w'. This method is naturally recursive and arrives to the solution of the problem by using the optimal solution of a sub-problem, thereby satisfying the constraints of a dynamic programming solution.

Group – C (Computationally Hard - 10 marks – approx 20 min)

8. There are k lists where each list consists of sorted numbers in non-increasing order. There is a total of n number of items over all the lists. Suggest an efficient algorithm to pick the top-k items, where k<<n, (i. e. report the largest k items) and explain the time complexity of the algorithm.

Hint: A hippy guy will solve it fast, err…, did I spell anything wrongly? ☺☺ (3 + 2 = 5)

Ans. Put the 1st element from each of the lists in a max-heap, with each element keep a tag to which list it belonged. It will take O(k) time as there are k lists. Now start extracting the max (topmost) element and take the next best element from the list where it originally belonged. Insert this new element in the heap. The best way to insert is put it as the 1st element in the array (used to implement the the heap) and then just call max-heapify on that position. In O(log k) time it will become max-heap again. Just repeat this whole operation k times. The time required will be O(k logk) { N. B. More precisely O(k + k.1 + (k – 1) log k) as the max-heapify call is not required for the last extraction}

9. Mr. Ambidex Bannerjee is a VLSI engineer. He wants to connect n circuit points to the clock signal. Now, the clock signal is going to pass parallel to the x-axis and all those circuit points are going to be connected by wires which are all vertical to the clock line. Please look at the adjoining figure to have a feel. Now, if the coordinates (x_i, y_i) for each circuit point c_i to be connected are given, how will you determine the optimal placement of the clock line so that the total wire-length L for connecting the circuit

HERITAGE INSTITUTE OF TECHNOLOGY

Class Test I Examination 2017 Session: 2016 - 2017

Discipline: Computer Science & Engineering

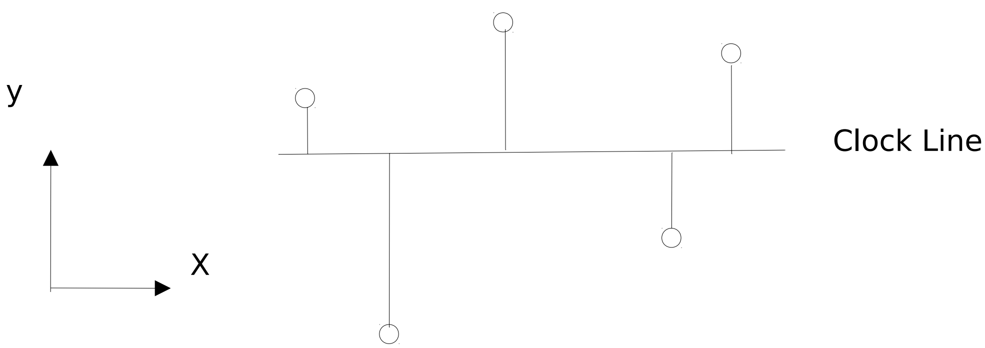
Paper Code: CSEN 2201 Paper Name: Design & Analysis of Algorithms Date: 8th March 2017

Time – 70 min Total Marks: 30

Total marks is 36 but the maximum you can score is only 30

points to the clock line is minimized. In other words, if the equation of the clock line is $\mathbf{y} = \mathbf{y_c}$ then what should be the value of $\mathbf{y_c}$ in order to minimize $\mathbf{L} = \sum_{i=1}^{n} (y_i - y_c)$. Justify your answer.

Hint: This question though long is not difficult. Can you guess this question is from which chapter? If you are stuck, remember INDUCT I ON. (5)



Ans: In one word, the answer is that the clock line should pass through that circuit point which has the median y-co-ordinate. (Obviously x co-ordinate does not matter). If n is even, however, it can pass through any of the two medians or though any y-co-ordinate between the two medians. If somebody writes median, please give him 1 mark.

Now, note carefully that I have not asked for any algorithm, I asked for the proof why median is the answer. So if someone write the algorithm for finding median he does not get any extra credit beyond 1 mark.

Justification – Combine the topmost and bottommost (in terms of y co-ordinate) circuit point. Look together they contribute the same length of metal wherever I put the clock line between them. So to find the answer I can as well ignore the pair. Once deleted from my consideration, I can then pair up the next topmost and next bottommost. Then apply the same logic and in this way if I am left with one point then the clock line must pass through that and that is indeed the median. If however, I am left with only 2 points (in case I start with even number of points), then I have already given the answer in the previous paragraph.

HERITAGE INSTITUTE OF TECHNOLOGY

Class Test I Examination 2017 Session: 2016 - 2017

Discipline: Computer Science & Engineering

Paper Code: CSEN 2201 Paper Name: Design & Analysis of Algorithms Date: 8th March 2017

Time – 70 min Total Marks: 30

Total marks is 36 but the maximum you can score is only 30

Look this can be proved very formally by induction separately on odd points and even points.

For one point, the clock line must pass through it. For 3 points it must pass through the middle point. If it it is true for 2k +1 points, i.e. the clock line should go through the median, then for (2k +1) + 2 points, I can simply argue that ignoring the two boundary points there are 2k +1 points within, whose answer is the median and then adding the two boundary points does not bias the answer at all. (Proved)

Likewise prove for even number of points.

N.B. The answer does not change at all even if the y co-ordinates are not all distinct.