CONFIDENTIAL

HERITAGE INSTITUTE OF TECHNOLOGY

Class test I / II / III Examination 2018 Session : 2017-2018

Discipline : B.Tech (CSE)

Paper Code : CSEN2201 Paper Name : Design & Analysis of Algorithms

Time Allotted : 1 hr Full Marks : 30

Figures out of the right margin indicate full marks.

Answer all questions.

Candidates are required to give answer in their own words as far as practicable.

Total marks is 35 but the maximum you can score is only 30.

---

1 (a) Suppose we have a O(n) time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use that median as a pivot. What will be the worst case time complexity of this modified QuickSort? 1x4=4

i) $O(n^2 \log n)$ ii) $O(n^2)$ iii) O(n logn logn) **iv)** O(nlogn)

(b) Let W(n) and A(n) denote respectively, the worst case and average case running time of an algorithm executed on an input of size n. which of the following is ALWAYS TRUE?

i) $A(n) = \Omega(W(n))$ ii) $A(n) = \theta(W(n))$ iii) $A(n) = O(W(n))$ iv) $A(n) = o(W(n))$

(c) Let T(n) be a function defined by the recurrence $T(n) = 2T(n/2) + \sqrt{n}$ for $n \geq 2$ and T(1) = 1 Which of the following statements is TRUE?

i) $T(n)= \theta(\log n)$ ii) $T(n)= \theta(\sqrt{n})$ iii) $T(n) = \theta(n)$ iv) $T(n) = \theta(n \log n)$

(d) In all pair shortest path problem solved by Floyd-Warshall algorithm, let $\pi[i,j]^{(k)}$ and $d[i,j]^{(k)}$ be respectively the typical element of predecessor matrix and distance matrix of iteration k. Then if $d[i,j]^{(k+1)} < d[i,j]^{(k)}$ then $\pi[i,j]^{(k+1)} =$

i) $\pi[i,j]^{(k)}$ ii) $\pi[i, k]^{(k+1)}$ iii) $\pi[k, j]^{(k-1)}$ iv) $\pi[k ,j]^{(k)}$ v) $\pi[j, k]^{(k)}$

This question seems to be wrong. The correct answer should be $\pi[k + 1, j]^{(k)}$ but there is no option with that answer. So marks should be given to all.

---

2 (a) Consider the following recurrence: (4)+(4)+(2) = 10

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

Show that the solution to the above recurrence is $\Omega(2^n)$

(Hint. You may use the substitution $T(n)>= 2^n$)

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

$T(n) \in \Omega(2^n)$ then $T(n) \geq c2^n$ (c is a constant, c > 0)

	P(n) = P(1) + P(n – 1) + P(2) + P(n - 2) + ...+ P(n - 2) + P(2) + P(n – 1) + P(1) = 2(P(1) + P(2) + ...+ P(n – 1))......Let the whole expression is X. Then $X \geq 2c\,(2^1 + 2^2 + ...+ 2^{n-1})$ $X \geq 2c\,(2(2^{n-1} - 1) / (2 - 1))$ $X \geq 4c\,(2^{n-1} - 1)$ for c = 1 and $n \geq 2$, $X \geq c2^n$, which proofs $T(n) \geq c2^n$, so $T(n) \in \Omega(2^n)$

(b)	Find an optimal Huffman code for the following set of frequencies using Greedy strategy, based on the first 8 Fibonacci numbers as follows:
	a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21
	Also Explain the strategy briefly.
	
	Huffman tree:
	<pre> 54
	    0/ \1
	   h 33
	          0 / \1
	          g 20
	                 / \
	                f 12
	                       / \
	                      e 7
	                            / \
	                           d 4
	                               / \
	                              c 2
	                                   / \
	                                  a b</pre>So in this way h: 0 g: 10 f: 110... And so on as left child coded with 0 and right child coded with 1.

(c)	Consider the following recurrence: T(n) = 2T(n/2) + log n, T(0) = 0 Is it possible to apply standard Master’s Theorem on the given recurrence to find the solution? If so apply the theorem and give the solution. If not, then give the proper reason for that. T(n) = 2T(n/2) + log n f(n) = logn, now obviously $f(n) \in O(n \wedge(\log_b^{a} - C)) = O(n \wedge (\log_2^{2} - C)) = O(n^{(1-}$

So $T(n) \in \Theta(\ n\ \wedge(\ \log_b{}^a\ )) = \Theta(n)$

3(a) Write the pseudo-code for finding out the longest common subsequence of two sequences X and Y of length m and n respectively. (4 + 3 + 2) = 9

Pseudo-code for LCS – array (m + 1) x (n + 1)

```
m <- length[X], n <- length[Y]
for i <- 1 to m
do c[i, 0] <- 0
for j <- 0 to n
do c[0, j] <- 0
for i <- 1 to m
do for j <- 1 to n
        do if x_i = y_j
            then c[i, j] <- c[i - 1, j – 1] + 1
                 b[i, j] <- " ↖"
          else if c[i - 1, j ] >= c[i, j – 1]
                 then c[i, j] <- c[i – 1, j]
                      b[i, j] <- "↑"
                 else c[i, j] <- c[i, j - 1]
                      b[i, j] <- "←"
Return c and b
```

(b) You are provided as input an integer $n > 1$ and a permutation $p_1 p_2 \ldots p_n$ of the integers $12\ldots n$. Describe **briefly** a method that will find the longest increasing subsequence (LIS) and also the longest decreasing subsequence (LDS) 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 and (8 7 4 2) is strictly decreasing also of 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, same holds true for the decreasing subsequence.

Ans. Just sort the numbers in O(n log n) time. Then just find the LCS of the original sequence and the sorted sequence in O(n^2) time to find the LIS and also find the LCS of original sequence and reverse ordered sorted sequence to get the LDS.

(c) Will |LIS| = |LDS| for any permutation sequence? If yes, prove it, if not, give the shortest possible counter-example.
Ans. No. eg. 1 2, |LIS| = 2, |LDS| = 1.

4. Explain when does the worst case for MAX-HEAPIFY occur? Also prove that the running time of BUILD-MAX-HEAP is tightly bounded by O(n). (3 + 3) = 6

CONFIDENTIAL

HERITAGE INSTITUTE OF TECHNOLOGY

Class test I / II / III Examination 2018 Session : 2017-2018

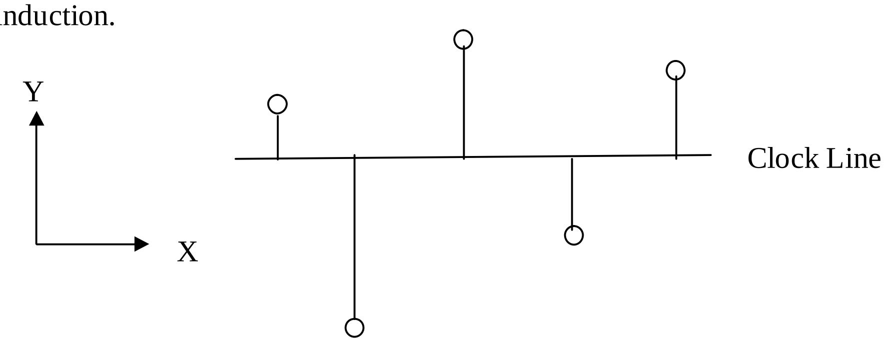
Discipline : B.Tech (CSE)

Paper Code : CSEN2201 Paper Name : Design & Analysis of Algorithms

Consult Cormen topic : 6.3

5. Mr. Nano is a ULSI 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 points to the clock line is minimized. Justify your answer. (3 + 3) = 6

Hint: Can you guess this question is from which chapter? For justification, think induction.



For odd n, the horizontal line has to pass exactly through the circuit point with median y-coordinate.

For even n, the horizontal line has to pass anywhere between the two circuit points with median y-coordinates. Remember for even n, there are two medians.

Proof – Basically if you just take 2 circuit points, then for the horizontal line that passes through anywhere between them, the sum of the vertical distances of those two points from the horizontal line remains invariant. Now extend this concept by pairing up 2 circuit points with max and min y-coordinates and then delete them from consideration as their deletion will not change the solution. Finally you will arrive at the solution stated above where either only one or two lines will remain.