B.TECH/CSE/3RD SEM/CSEN 2101/2018
DATA STRUCTURE & ALGORITHMS
(CSEN 2101)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and
any 5 (five) from Group B to E, taking at least one from each group.

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

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) Two matrices, with number of non-zero elements k1 and k2 respectively, have been stored in their sparse representation (triple format) in sorted order of the row and column indices, respectively. Worst-case time complexity for adding the matrices, can be performed in
(a) O(min(k1, k2)) (b) O(max(k1, k2))
(c) O(k1 + k2) (d) O(k1 * k2).

(ii) Which data structure should be used for implementing an iterative algorithm of pre-order traversal?
(a) Stack (b) Input and output restricted deque
(c) Queue (d) None of the above.

(iii) If a graph has 110 nodes and 98 edges, what would be the minimum number of components it may have?
(a) 10 (b) 96 (c) 12 (d) 95.

(iv) Which of the following trees are guaranteed to be balanced?
(a) AVL tree, almost-complete binary tree (b) B-Tree, BST
(c) 2-tree, almost-complete binary tree (d) AVL tree, 2-tree.

(v) Suppose you are applying radix sort on n number of b-bit binary numbers organized as r-bit digits; the time complexity will be
(a) $O(n/r * (n + 2^r))$ (b) $O(b/r * (n + 2^r))$
(c) $O(b/r * (b + 2^r))$ (b) $O(b/r * (n + 2^{b/r}))$.

(vi) Which of the following statements is true?
(a) The time complexity of interpolation search is always better than binary search.
(b) The time complexity of binary search is always better than interpolation search.
(c) The worst-case complexity of interpolation search is better than worst-case complexity of binary search
(d) The average case complexity of interpolation search is better than average case complexity of binary search.

(vii) Which of the following is true?
(a) The cost of searching an AVL tree is θ(log n) but that of a binary search tree is O(n)
(b) The cost of searching an AVL tree is θ(log n) but that of a complete binary tree is θ(n log n)
(c) The cost of searching a binary search tree is O(log n) but that of an AVL tree is θ(n)
(d) The cost of searching an AVL tree is θ(n log n) but that of a binary search tree is O(n).

(viii) Suppose you place m items in a hash table with an array size of s. What is the correct formula for the load factor?
(a) s + m (b) s - m (c) m * s (d) m / s.

(ix) In an in-orderly threaded binary tree, an internal node having no right child, will contain address of ______________ in its right child address field.
(a) in-order successor node (b) in-order predecessor node
(c) root node (d) itself.

(x) If a full 3-ary tree T contains 6 internal nodes, then what will be the total number of nodes present in the tree?
(a) 20 (b) 14 (c) 19 (d) 18

Group – B

2. (a) Find the sparse representation (triple format) for the following matrix. Check whether it is useful to use the sparse representation instead of the original matrix.

$$\begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Given a matrix of dimensions m × n, prove that its sparse representation (triple format) will be more space efficient than the original matrix only if the number of non-zero elements is approximately bounded by one-third of the total number of elements that can be stored in the matrix.

(c) Find the Big-Oh for $T(n) = 48n^{100} + 2^{n+2} + 3n^2 + 100$

(4 + 2) + 4 + 2 = 12

3. (a) Write a function in C/pseudo-code to delete the node with the minimum value, from a singly linked list. If there are multiple occurrences, delete only the first occurrence. Discuss the time complexity of the algorithm.

(b) Fill up the table with the worst-case time complexity for the respective functions –

Type of linked list	Insert at front	Insert at end	Delete from front	Delete from end
SLL with tail	O(1)		O(1)	
DLL with tail	O(1)			
CSLL				O(n)
CSLL with tail				

LL = Linked List, S = Single, D = Double and C = Circular

6 + (0.5 × 12) = 12

Group – C

4. (a) Write an algorithm to check for balanced parentheses in an expression.

(b) Convert the given infix expression to postfix expression
4 * (40 + 7) / (3 * (11 - 4))
State what type of stack or stacks will be required to perform this operation, also show the content of the stack or stacks.

(c) Determine the time complexity of the recursive solution of the "Tower of Hanoi" problem.

6 + 3 + 3 = 12

5. (a) Solve the "Tower of Hanoi" problem using recursion. Write the C-code or pseudo-code.

(b) Explain the disadvantages of backtracking.

(c) What is the purpose of tail recursion?

(d) Write a tail recursive function in C to calculate the sum of first n numbers (up to Fib(n)) in Fibonacci series. (Consider Fib(0) = 0)

3 + 2 + 2 + 5 = 12

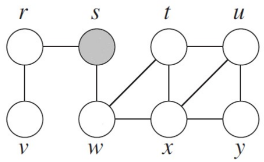
Group – D

6. (a) Suggest an algorithm to check whether a given tree is a binary search tree or not?

(b) Consider the following sequence of keys 40,15,65,35,55,45,75,95,85,05,30, which will be used to create an AVL tree. Show the intermediate steps for how this AVL tree will be built.

7 + 5 = 12

7. (a) Perform depth-first traversal on the graph in the given figure, with s as the starting node. Draw the depth-first tree, mark the tree edges with solid lines and mark the non-tree edges with dotted lines.



(b) (i) Prove that in a 2-tree (i.e. a full binary tree), if the sum of all external paths is E, the sum of all internal paths is I and Q is the total number of non-leaf nodes, then E = I + 2Q.

(ii) Prove that the sum of degrees of all nodes equals 2 times the number of edges in a graph.
Hint: You may try to use **Induction**. Some other method is also fine.

6 + (4 + 2) = 12

Group – E

8. (a) Consider below the pseudo-code that is very similar to *Improved Bubble Sort* method that was taught in class. There may be one or more mistakes introduced somewhere in this code. Can you identify in which lines the mistakes are? Also suggest the necessary correction so that it performs the intended optimization. You need not copy the whole code; just write the corrected portion along with line number.

```
1.  Bound ← n
2.  do
3.  {
4.              t ← 0
5.              for j = 1 to BOUND -1
6.                      if key(j) > key(j + 1)
7.                          interchange record(j) ↔ record(j + 1)
8.               end if
9.               t← j
10.             end for
11.             BOUND ← t
12. } while ( t >= 0)
```

(b) Write the pseudo-code for insertion sort. State its best-case, worst-case and average-case asymptotic complexity.

5 + (5 + 2) = 12

9. (a) State the advantage and disadvantage of interpolation search over binary search.

(b) Propose an algorithm for finding the first non-repeated character in a string containing only lowercase English characters, e.g. , if the string is "abzdabb" the answer is 'z', and then also analyze its time complexity.
Hint: Try to solve it in at most 2 passes.

(c) Explain each step of the radix sort algorithm for the given set of data:
20, 2, 183, 25, 15, 105, 40, 21, 9, and 58.

4 + 5 + 3 = 12