



## AUTUMN END SEMESTER EXAMINATION-2017

3<sup>rd</sup> Semester B.Tech & B.Tech Dual Degree

### DATA STRUCTURE AND ALGORITHM

#### CS-2001 / CS-301

(Regular-2016 & Back of Previous Admitted Batches)

Time: 3 Hours

Full Marks: 60

**Answer any SIX questions including question No.1 which is compulsory.**

*The figures in the margin indicate full marks.*

*Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.*

1. Answer all.

[2 × 10]

(a) What is the time complexity of following function fun()?

```
void func() {  
    int i, j;  
    for (i=1; i<=n; i++)  
        for (j=1; j<=log(i); j++)  
            printf("KIIT University");  
}
```

(b) Write a pseudo code to search an element in a single linked list.

(c) Distinguish between tridiagonal & triangular sparse matrix with example.

(d) Write a function to test whether a loop exists in a single linked list.

(e) What is stack ADT error & implementation error? Explain with suitable example.

- (f) Let *ptr* is a pointer pointing to a node in a circular single linked list. Write pseudo code to delete the node to which *ptr* is pointing.
- (g) Write the condition for overflow and underflow situation in a circular queue.
- (h) Find the preorder traversal of a binary tree which has been constructed from the given postorder and inorder traversals:  
in[] = {4, 8, 2, 5, 1, 6, 3, 7}, post[] = {8, 4, 5, 2, 6, 7, 3, 1}.
- (i) Write a C code to find number of nodes in a binary tree.
- (j) Construct a max heap with the following data elements.

25, 30, 15, 8, 27, 22, 12, 7, 18

- 2. (a) Write a algorithm/ function to find the minimum difference [4]  
between any two elements in a given array.
- (b) Write an algorithm/ pseudo code/ C code snippet to remove [4]  
duplicate elements from an unsorted single linked list.
- 3. (a) Write pseudo code to reverse a single linked list while [4]  
traversing the list once.
- (b) Write an algorithm/ pseudo code to merge two sorted linked [4]  
lists to produce a sorted linked list.
- 4. (a) Write C -pseudo code to implement the push and pop [4]  
operations of two different stacks at two ends of a given array.
- (b) Write C -pseudo code for evaluation of postfix expression [4]  
using stack.

5. (a) Why AVL trees are used? Discuss the all steps to ensure that the given tree remains AVL after every insertion. Illustrate the steps involved while building an AVL tree with values: 15, 20, 24, 10, 13, 7, 30, 36, 25. [6]
- (b) Explain recursive in order tree traversal with suitable example. [2]
6. Write the pseudo code for quicksort including partition() function. Also illustrate the step-by-step process of partition() function with a given array: {10, 80, 30, 90, 40, 50, 70}, considering last element as the pivot element. [8]
7. (a) Differentiate between BFS and DFS with suitable example. [4]
- (b) What do you mean by collision in hashing? How to handle Collisions? Draw the neat diagram to illustrate separate chaining method to handle collisions using hash function as "key mod 7" and sequence of keys as 50, 700, 76, 85, 92, 73, 101. [4]
8. (a) Define B-tree. Create a B-tree of order 3 with the following elements. [4]
- 8, 9, 10, 11, 12, 7, 6, 5, 20, 26, 22
- (b) What is threaded binary tree? What are its advantages? Explain with an example of a in order threaded binary tree. [4]

- \*\*\*\*\* -