| U.S.N. | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|

# B.M.S. College of Engineering, Bengaluru-560019

**Autonomous Institute Affiliated to VTU**

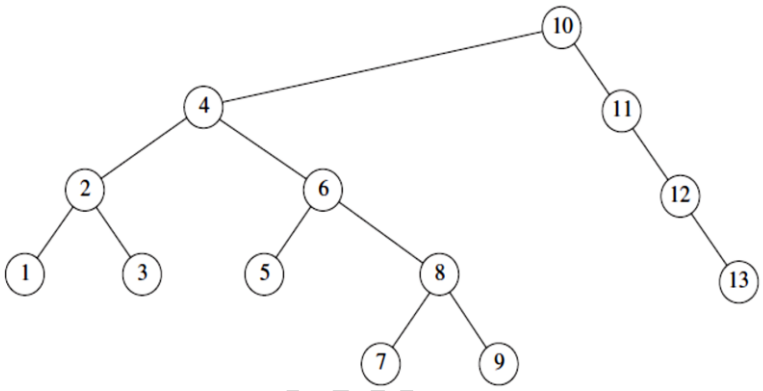
## January / February 2025 Semester End Main Examinations

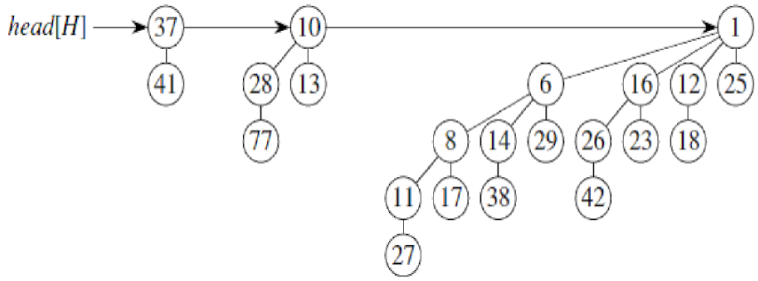
**Programme: B.E.** **Semester: VI**
**Branch: Computer Science and Engineering** **Duration: 3 hrs.**
**Course Code: 22CS6PEADS** **Max Marks: 100**
**Course: Advanced Data Structures**

**Instructions**: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

**Important Note:** Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

| | | UNIT - I | CO | PO | Marks |
|---|---|---|---|---|---|
| 1 | a) | Show the result of the following sequence of instructions: union (1,2), union(3,4), union(3,5), union(1,7), union(3,6), union(8,9), union(1,8), union(3,10),union (3,11), union(3,12), union(3,13), union(14,15), union(16,0), union(14,16),union (1,3), union(1, 14) when the unions are performed by height. | CO1 | PO1 | 6 |
| | b) | List advantages and disadvantages for the following self-organizing lists:<br>**i)** Move to Front Method<br>**ii)** Transpose Method | CO2 | PO2 | 6 |
| | c) | Write a C code for implementation of memory efficient doubly linked list. | CO1 | PO1 | 8 |
| | | **OR** | | | |
| 2 | a) | Demonstrate and show the contents of the Skip list after inserting the following sequence of elements:<br>➢ **21 with level 0**<br>➢ **43 with level 3**<br>➢ **76 with level 0**<br>➢ **16 with level 2**<br>➢ **26 with level 0**<br>➢ **251 with level 2**<br>➢ **98 with level 3**<br>➢ **87 with level 0**<br>➢ **79 with level 1**<br>After the skip list is constructed, perform the following deletes and update the list. Show steps clearly.<br>**I. Delete 16**<br>**II. Delete 79** | CO1 | PO1 | 10 |
| | b) | Demonstrate the usage of Unrolled linked list. List its advantages and disadvantages. | CO1 | PO1 | 10 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | **UNIT - II** | | | |
| | 3 | a) | Show the AVL tree that results after each of the integer keys: 9, 27, 50, 15, 2, 21, and 36 are inserted, in that order, into an initially empty AVL tree. | ***CO1*** | ***PO1*** | **10** |
| | | b) | Show that every AVL tree can be colored as a Red-Black tree. Are all Red-Black trees AVL? | ***CO2*** | ***PO2*** | **4** |
| | | c) | List the properties of 2-3 trees with example. | ***CO1*** | ***PO1*** | **6** |
| | | | **OR** | | | |
| | 4 | a) | What is the significance of the 'α' parameter in a Scapegoat Tree? How does it affect the tree's structure and performance? | ***CO1*** | ***PO1*** | **4** |
| | | b) | Show the result of accessing the key 5 in order in the Splay tree in below Figure. Analyse how splay tree is better in practical situation.   | ***CO2*** | ***PO2*** | **8** |
| | | c) | Demonstrate with suitable example various rotations on Red-Black trees. | ***CO3*** | ***PO3*** | **8** |
| | | | **UNIT - III** | | | |
| | 5 | a) | Analyze and compare the efficiency of Tries, Suffix tree and Suffix arrays. | ***CO2*** | ***PO2*** | **6** |
| | | b) | Draw the Suffix tree for S = ABAAB$. | ***CO1*** | ***PO2*** | **7** |
| | | c) | Given an array A=[1,3,5,7,9,11], design a segment tree to support efficient range sum queries. | ***CO1*** | ***PO1*** | **7** |
| | | | **OR** | | | |
| | 6 | a) | Construct a Fenwick Tree for the following array {3,2,-1,5,7,-3,2,1}. Also show how the sum from 0 to 6 is computed. | ***CO1*** | ***PO1*** | **9** |
| | | b) | Consider following points in a 2-D plane: (3, 6), (17, 15), (13, 15), (6, 12), (9, 1), (2, 7), (10, 19), construct a K-D tree. | ***CO1*** | ***PO1*** | **7** |
| | | c) | Compare the efficiency of finding the minimum value in a k-d tree versus a balanced binary search tree. | ***CO2*** | ***PO2*** | **4** |
| | | | **UNIT - IV** | | | |
| | 7 | a) | Construct a Hash table for the following numbers {20, 50, 53,75, 100, 67, 105, 3, 36, 39} using Cuckoo hashing technique. | ***CO1*** | ***PO1*** | **10** |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | b) | What strategies are used to maintain the neighborhood property during insertions and deletions? Analyze the efficiency of these strategies in minimizing collisions and maintaining performance. | *CO2* | *PO2* | **10** |
| | | | **OR** | | | |
| | 8 | a) | Demonstrate Perfect hashing with an example. | *CO1* | *PO1* | **10** |
| | | b) | Apply Hopscotch hashing and construct the hash table for the following set of elements:<br>{151,61,23,47,189,79,101,12,31,145,89,132}.<br>Assume neighborhood=4. List the advantages of Hopscotch hashing. | *CO1* | *PO1* | **10** |
| | | | **UNIT - V** | | | |
| | 9 | a) | Discuss the properties of Binomial Tree. | *CO1* | *PO1* | **6** |
| | | b) | Show the result of performing three ExtractMin operations in the Binomial heap in the below figure:<br>  | *CO1* | *PO1* | **10** |
| | | c) | Can both insert and findMin be implemented in constant time in Binomial Heap? Justify. | *CO2* | *PO2* | **4** |
| | | | **OR** | | | |
| | 10 | a) | Demonstrate the various operations performed on Binomial Heap with an example. | *CO3* | *PO1* | **10** |
| | | b) | Explain with an example Decreasing key operation in the Fibonacci Heap with an example. | *CO3* | *PO1* | **10** |

**********