|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **P.V.P Siddhartha Institute of Technology** | | | | | | | | | |
| **Department of Computer Science and Engineering** | | | | | | | | | |
| **Course: B.Tech** | | **Year: II** | **Semester: I** | **Descriptive: I** | **A.Y:2024-25** | | | | |
| **Subject Code:23CS3301** | | **Subject Name: Advanced Data Structures and Algorithm Analysis** | | | **Regulation:PVP23** | | | | |
| **Duration:1 hr 30 min** | | **Maximum Marks:30Marks** | | | **Date:19/09/24** | | **Session: F.N** | | |
| **Answer one from each Either-Or type of Questions.**  **Each Question carries 10 Marks**  **3×10M=30M** | | | | | | | | | |
|  | | | | | | | | | |
| **Q.No** |  | | | | | **Marks** | | **CO** | **Level** |
| **1.** | Construct an AVL tree for the given elements 9, 2, 5, 7 ,3, 1, 4, 6, into an initial empty tree. Delete 7, 5, 9, 6,9 from the build AVL tree. And again insert 65, 70, 22, 55,13, 79, to the resultant tree. | | | | | **10** | | **CO3** | **L3** |
| **(OR)** | | | | | | | | | |
| **2.** | Apply insertion operation on B - Tree of order 3 for the given elements 4, , 6, 19, 12, 29, 32, 27, 22, 3, 33, 89, 26, 13, 15, 7, 16, 49. | | | | | **10** | | **CO3** | **L3** |
|  | | | | | | | | | |
| **3.** | a) Build insertion algorithm for max heap | | | | | **5** | | **CO3** | **L3** |
| b) Develop a max heap for the given elements 142, 543, 123, 65, 453, 879, 572, 434, 111, 242, 811, 102. | | | | | **5** | | **CO3** | **L3** |
| **(OR)** | | | | | | | | | |
| **4.** | Trace the Breadth First Traversal & Depth First Traversal for the given graph as shown in below **Figure**. And also construct the BFS, DFS spanning tree. Consider the start vertex as 1**.** | | | | | **10** | | **CO3** | **L3** |
|  | | | | | | | | | |
| **5.** | Identify Big-oh (O), Omega (Ω) and Theta (ϴ ) representation of given function f(n) = n3 2 n + 6n2 3 n . Justify your answer. | | | | | **10** | | **CO3** | **L3** |
| **(OR)** | | | | | | | | | |
| **6.** | a) Build BFS and DFS graph traversal algorithms using Queue and Stack linear data structures | | | | | **10** | | **CO3** | **L3** |