CP-CS1-M

Binary Search Trees | Heap | Hashing | Disjoint Sets

0. **Implement CRUD in BST (Medium)**
1. **Implement CRUD in Heap (Medium)** => Priority Queue
2. H/W: Implement Heap Sort (Medium)
3. Construct BST from preorder traversal (East to Medium)
4. Median of a stream of running integers (Hard)
5. **Merge K Sorted Arrays (Medium - Hard)**
6. Kth Largest/Smallest Element in an array (Hard)
7. **Largest BST in Binary Tree (Hard)**
8. LCA of BST (Easy)
9. Inorder Successor in BST (Medium)
10. **Sorted Array to BST (Easy)**
11. Given n appointments, find all conflicting appointments (Hard) / Let's not talk about this as this question is the application of Interval Trees which is internally avl tree.
12. Find kth smallest element in BST (Order Statistics in BST) (Medium)
13. Construct BST from its given level order traversal (Hard)
14. Print BST keys in the given range (Easy)

- Talk about hashing algorithms
    - Implement Map (Ordered/Unordered Map)