

Review Sheet for Test 2
CSCI 340 Fall 2023

Lambda expression and High order function (used in assignment 4)

STL map (used in assignment 5)

Recursion (used in many tree operations)

Tree/Binary Tree/BST

- Tree.
 - What is a tree?
 - What is a binary tree?
 - What is a complete binary tree?
 - What are the properties of a tree? -- Do you know the terms such as leaf nodes, height, node level, and paths?
 - What is a binary search tree?
- Binary Tree and its major operations
 - Traversal: preorder, inorder, postorder, level-order.
 - Get tree height
 - Get node count
 - Get number of leaves
 - Delete a binary tree
 - How to implement the above operations (using recursion if applicable)?
- Binary Search Tree and its major operations
 - Find
 - Insertion
 - Deletion: leaf, one child, two children
 - How to find a predecessor or a successor?
 - Deletion by copying vs deletion by merging.
- Operation complexity.

AVL tree:

- Definition of an AVL tree. What is the balance factor of a node?
- Why use AVL tree (instead of using BST)?
- Insertion:
 - 4 cases to re-balance the AVL tree.
 - single rotation versus double rotation
 - How to build an AVL tree using insertion (and rotation when needed)?
- Deletion: Why there can be a need of propagation?
- Pros and cons of AVL trees.
 - E.g. All operations guaranteed $O(\log N)$.

Note: You need to be able to draw the contents of the tree during and after the AVL insertion operation is performed.

Heap

- Basic definitions of a heap data structure
 - What is a heap? What is the heap-order property?
- Why use heap to implement a priority queue? (Hint: Efficiency).
- How to enqueue (insert)?
- How to dequeue (deleteMin or deleteMax)?
- How to build a heap?
 - Bottom-up approach -- heapify!
 - top down approach
 - What are their complexity? Which approach is a linear algorithm?
- What does moveDown() do?
 - Where is it used?
 - How to implement it using an array?
- Heap sort (Not part of Test 2)
 - How does it work?
 - Complexity: worst case: $O(N \lg N)$; best case: $O(N)$.

You need to be able to draw the contents of the array or the binary tree during and after the heap operations are performed, e.g. how to heapify a given array.

Assignments:

Assignment 4: Lambda and STL high order function

Assignment 5: map and string

Assignment 6: binary tree

Assignment 7: binary search tree:

- preorder(root, fn)
- inorder(root, fn)
- postorder(root, fn)
- levelorder(root, fn)
- delete_tree(root)
- height(node)
- count(root)
- bst_find(root, value)
- bst_insert(root, value)
- bst_remove_value(root, value)
- is_bst(root)
- bst_minimum(root)
- bst_maximum(root)
- successor(node)