CS251 - Data Structures and Algorithms
Fall 2024

PSO 7, Week 8

Question 1

(Insertion and Deletion)

(1) Insert $\{15, 21, 7, 24, 0, 26, 3, 28, 29\}$ into an initially empty Left–Leaning Red–Black tree.

(2) Delete 7 in the final Left–Leaning Red–Black tree obtained in question (1).

Question 2

(LLRB tree)

(1) Is it true that the maximum key in a LLRB-tree always appears at a leaf node.

(2) Given a set of keys $\{1, 2, 3, 4, 5, 6, 7\}$, insertion in what order to an initially empty LLRB-tree will be free of rotation (i.e. no rotation is required)?

(3) Given $\{1, 2, 3, 4, 5, 6, 7\}$, what is the maximum number of red nodes of a valid LLRB-tree containing these values? Insertion in what order to an initially empty LLRB-tree will result in such a tree?

Question 3

(True or False)

(1) 2-3 trees are always perfectly balanced.

(2) The maximum height of any 2-3 tree with n keys is at most $O(\log_2(n))$.

(3) The maximum height of any 2-3-4 tree with n keys is at most $O(\log_2(n))$.

(4) The sequence $\{3, 1, 5, 7, 9, 13, 11\}$ can represent an inorder traversal of a valid Red-Black Tree.

(5) The maximum number of rotations required for insertion in a LLRB tree is $\Theta(\log n)$.

Question 4

(Binary search tree)

Find the k^{th} smallest element in a BST in $O(h)$ time, where h is the height of the tree. (You are allowed to modify the BST for free.)

Question 5

(Adjacency-list Representation)

1. Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of a vertex? How long does it take to compute the in-degree of a vertex?

2. The transpose of a directed graph $G = (V, E)$ is the graph $G^\top = (V, E^\top)$, where $E^\top := \{(v, u) : (u, v) \in E\}$. In other words, $G^\top$ is G with all its edges reversed. Describe an efficient algorithm for computing $G^\top$ from G for the adjacency-list representations of G and analyze the runtime of your algorithm.

3. The square of a directed graph $G = (V, E)$ is the graph $G^2 = (V, E^2)$, where $(u, v) \in E^2$ if and only if G contains a path with at most two edges between u and v. Describe an efficient algorithm for computing G^2 from G for the adjacency-list representations of G and analyze the runtime of your algorithm.

Question 6

(Adjacency-matrix Representation)

1. Give an adjacency-matrix representation for a complete binary search tree on 7 vertices numbered from 1 to 7.

2. Show how to determine in $O(|V|)$ time, whether a directed graph G contains a **universal-sink**, i.e. a vertex with in-degree $|V|-1$ and out-degree 0, given an adjacency-matrix for G.