CIS 121—Data Structures and Algorithms—Spring 2020

Tries and Binary Search Trees—Monday, April 20 / Tuesday, April 21

Readings

- Lecture Notes Chapter 24: Tries
- Lecture Notes Chapter 25: Balanced Binary Search Trees

Problems

Problem 1
Given a set of $N$ strings, how can we find the longest common prefix between any two strings? Analyze the running time of your algorithm and justify its correctness.

Problem 2
Given some string $S$, how can we find the longest repeated substring? What about if we want the longest that is repeated $k$ times?

Problem 3
Design an algorithm to decide if a given binary tree is a valid binary search tree.

Problem 4
Perform a series of rotations to make 5 the root of the following unbalanced BST:

```
    10
   / \
  7   15
   / \
  5
   / \
 3   6
 / \
2   4
 \ 
 1
```

Problem 5
Draw the resulting AVL tree after inserting the following:

```
{3, 4, 5, 8, 12, 17, 1}
```

Draw the resulting tree after deleting 12.
Additional Problems

Problem 6

Show that any arbitrary $n$-node binary search tree can be transformed into any other arbitrary $n$-node binary search tree using $O(n)$ rotations. Assume that both trees contain the same $n$ distinct elements. (Hint: First show that at most $n - 1$ right rotations suffice to transform the tree into a right-going chain, which is a binary search tree where every internal node only has a right child.)