CSC 225 SPRING 2018
ALGORITHMS AND DATA STRUCTURES I
ASSIGNMENT 3
UNIVERSITY OF VICTORIA

1. Insert items with the following keys (in the given order) into an initially empty binary search tree: 30, 40, 24, 58, 48, 26, 11, 13. Compute the tree after each insertion output the tree at the end of all the insertions.

2. Prove that no comparison-based algorithm can build a BST on N nodes using fewer than $\log N!$ comparisons.

3. Let D be a dictionary ADT with N items implemented with a AVL tree. Show how to implement the following method $\mathit{countAllInRange}(k_1, k_2)$ for D. This method should compute and return athe number of keys in D such that $k_1 \leq k \leq k_2$. Your algorithm should run in time $O(\log n)$.

4. Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 and 5, using the hash function $h(i) = (2i + 5) \pmod{11}$. Assume that collisions are handled by linear probing.

5. Suppose we use double hashing to resolve collisons; that is, we use the hash function $h(k; i) = (h_1(k) + ih_2(k)) \pmod{t}$. Show that if t is even and $h_2(k)$ is even for some key k, then the probe sequence for k examines at most half the slots in the table before returning to slot $h_1(k)$.