1. (5 + 15 = 20 points) Suppose there is a sequence of numbers, where the first 7 are \{8, 10, 12, 7, 3, 9, 2\}. Please show (1) the simple Binary Search Tree (without requiring balance) and (2) AVL-tree for these 7 numbers.

2. (15 \times 2 = 30 points) Let \( A = \{9, 11, 10, 5, 4\} \), show the hash tables (with the size \( m = 5 \)) for these 5 numbers:
   1. For chaining hash table, use the hash function \( h(x) = x \mod 5 \).
   2. For open addressing hash table, use the hash function \( h(x, i) = (x + i \cdot h'(x)) \mod 5 \), where \( h'(x) = 1 + (x \mod 3) \). The variable \( i \) counts the number of trials from 0 to \( m-1 \).

3. (15 \times 2 = 30 points) Following Problem 2, we continue to input another 5 numbers \{21, 1, 15, 32, 13\}, and double the sizes of the hash tables (since 10 is not a prime number, we update \( m \) to be 11). Please show the new hash tables:
   1. For chaining hash table, use the hash function \( h(x) = x \mod 11 \).
   2. For open addressing hash table, use the hash function \( h(x, i) = (x + i \cdot h'(x)) \mod 11 \), where \( h'(x) = 1 + (x \mod 3) \). The variable \( i \) counts the number of trials from 0 to \( m-1 \).

4. (20 points) In Problem 2 & 3, we set \( h'(x) = 1 + (x \mod 3) \) for open addressing hash table. Can we simply set \( h'(x) = x \mod 3 \) instead? Why?

5. (optional) Simulate the four query algorithms: simple BST, AVL-Tree, chaining hash table, and open addressing hash table. Compare their performances on building time and query time in practice. You can randomly generate a large sequence of integers to test.