EL9343 Homework 6

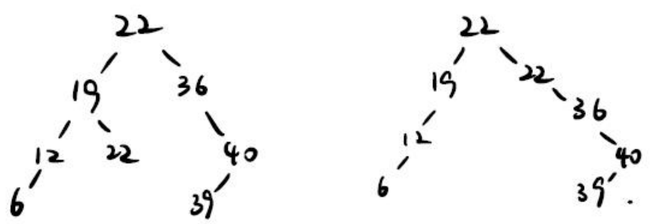
Due: Oct. 19th 8:00 a.m.

1. Let the table have 9 slots, and let the hash function be $h(k) = k \mod 9$. Demonstrate what happens when we insert the keys 10, 22, 35, 12, 1, 21, 6, 15, 36, 33 into a hash table with collisions resolved by chaining.
Solution:

$h(k)$	keys
0	36
1	$1 \to 10$
2	
3	$21 \to 12$
4	22
5	
6	$33 \to 15 \to 6$
7	
8	35

2. Suppose we use a hash function h to hash n distinct keys into an array T of length m. Assuming simple uniform hashing, what is the expected number of collisions?
Solution:
Suppose the keys are $k_1, k_2, \ldots, k_n$. Let X_i be the number of $j > i$ with $h(k_j) = h(k_i)$. Under the assumption of simple uniform hashing, $\mathbb{E}(X_i) = \sum_{j>i} \Pr(h(k_j) = h(k_i)) = \sum_{j>i} \frac{1}{m} = \frac{n-i}{m}$. Then, by linearity of expectation, the number of collisions C is the sum of X_i for all i, which makes $\mathbb{E}(C) = \mathbb{E}(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbb{E}(X_i) = \sum_{i=1}^{n} \frac{n-i}{m} = \frac{n^2-n}{2m}$.

3. (a) Given two BSTs, one in pre-order as $\{22, 19, 12, 6, 22, 36, 40, 39\}$, the other in post-order as $\{6, 12, 22, 19, 39, 40, 36, 22\}$, determine if the two BSTs are the same, not the same, or more information is needed.
Solution:
More information is needed, because there are two possible BSTs that have the same pre-order traversal as in the question, shown below:



The post-order sequence could only be generated from the left BST, not from the right one. Since we don't know which BST generates the pre-order traversal sequence, we can't decide on whether the two BSTs are the same.

(b) If all the keys in a BST are distinct, will the pre-order sequence result in an unique BST? If yes, please describe why; if no, find a counter-case.
Solution:
Yes. Given that the keys are distinct, there will not be occasions (like in 3a) when an element equals to the current root node and it can be placed in the left sub-tree as well as in the right sub-tree.
Therefore, an algorithm could be designed: for a sequence (or sub-sequence), the first element is the value of the root node. Then we should be able to find an unique point, where all the elements ahead of it are smaller than the root while all the elements beyond it are larger. (If such point doesn't exist, this sequence cannot be pre-order traversal of a BST.) All the elements between the first and this point forms a sub-sequence, and the resulting BST from this sub-sequence is the left sub-tree of the root. All the elements between this point and the end forms a sub-sequence, and the resulting BST from this sub-sequence is the right sub-tree of the root. Thus the BST we find is unique.
Note 1: This statement is also true if we substitute "pre-order" with "post-order".

Note 2: The given condition that all the keys are distinct is actually a very strong one. The post-order traversal sequence given in 3a could also lead to an unique BST. It is whether the separation point in the algorithm described above is unique or not.

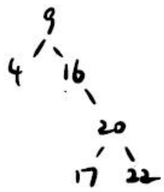
4. For the binary search tree (BST) in pre-order as $\{8, 4, 16, 9, 19, 17, 22\}$. **Please first draw the BST**, then show the result of following operations (each operation is carried out on the result of the previous operation):
   (a) Insert key 20;
   (b) Then, delete key 8;
   (c) Then, delete key 19;
   (d) Finally, delete key 16.

   Solution:

   Original:
   

   After a:
   

   After b:
   

   After c:
   

   After d: