Problem Set 3

Due date: Email your solutions to cs140_17fall@163.com or submit a written copy to a TA before 7:30 PM, November 2, 2017.

Problem 3-1. The chocolate lover [30 points]

Andy is organizing a chocolate-lovers party. He needs to buy at least w ounces of chocolate for the party. In the store, there are n chocolate bags each with the weight w_i (and they are not necessarily sorted based on the weights). Andy wants to buy the smallest number of bags. Provide an algorithm with running time $O(n)$ to help Andy.

Hint: Think about how the median might be helpful.

Problem 3-2. Binary Search Tree Practice [30 points]

(a) [10 points] **Traversal:** A common operation on a binary search tree is to print its elements in sorted order. This can be done using an *in-order traversal*: first find and print the minimum element of the tree (e.g. with $find\_min$), and then repeatedly find and print the next element (e.g. with $find\_next$). Because $find\_min$ and $find\_next$ each take $O(h)$ time, where h is the height of the tree, one can naively calculate a $O(nh)$ upper bound on the time to print all n elements using in-order traversal. Show that in fact, in-order traversal requires at most $O(n)$ time.

(b) [15 points] **Property Checking**: Given the root node of a binary tree, describe $O(n)$ time algorithms that evaluate whether the tree satisfies each of the following properties. Assume for this problem that the only data stored at a node is the node's key, and pointers to its parent, left child, and right child.

1. **BST Property:** the key of every node is greater than or equal to every key in the node's left subtree, and less than or equal to every key in the node's right subtree.
2. **AVL Property:** every node's left and right subtrees differ in height by at most one.

(c) [5 points] **Make AVL**: Consider the worst case binary search tree with no branching containing the keys $\{1, 2, 3, 4, 5, 6, 7\}$, with key 7 at the root. Show how to transform it into a binary search tree satisfying the AVL Property by performing a sequence of left and/or right rotations.

Problem 3-3. Strongly 2-Universal Hashing [40 points]

Let H be a family of hash functions, where each hash function $h \in H$ maps keys from the universe U to $\{0, 1, \ldots, m-1\}$. We call H **strongly 2-universal** if for all $x_1, x_2 \in U$ such that $x_1 \neq x_2$ and for a uniformly randomly chosen h, the pair $(h(x_1), h(x_2))$ is equally likely to be any of the m^2 pairs(y_1, y_2), where y_1 and y_2 are both elements of $\{0, 1, \ldots, m-1\}$.

(a) [10 points] Prove that if H is strongly 2-universal, then it is also universal.

(b) [10 points] Provide a hash family H that is universal but is not strongly 2-universal. Write your answer as a table, where each row corresponds to a hash function and each column corresponds to a key. For simplicity, try to make the values $|H|$, $|U|$, and m as small as possible.

Hint: You can find an answer where $|H|$, $|U|$, and m are at most 3.

(c) [10 points] Suppose we are given a universal hash family H. An adversary wants to force a collision, and the adversary knows exactly what H is, that is, the adversary knows the descriptions of each individual hash function $h \in H$. In particular, the following steps take place, in order.

1. We choose a hash function h from H uniformly at random. The adversary does not know which h we pick.
2. The adversary picks a key x and we tell the adversary $h(x)$.
3. The adversary picks a key y in an attempt to force a collision.

The adversary **succeeds** if in step 3 the adversary can choose a key $y \neq x$ such that $Pr[h(x) = h(y)] > 1/m$ (using the knowledge of the value of $h(x)$, but without knowing h).

Describe a universal hash family H for a general m such that the adversary can **succeed** on step 3 above. You may verbally describe your hash family (clearly and concisely), or adopt the the same format as in part (b) - however, note that in contrast with part (b), your table should scale to arbitrarily large m.

Note: You may write down H for a specific value of m to illustrate your idea, but you must clearly explain how to extend your construction to an arbitrarily large m.

(d) [10 points] Now consider the same set of steps in part (c), except with the assumption that H is strongly 2-universal. Show that the adversary cannot succeed, i.e. force a collision with probability greater than $1/m$ in step 3, on *any* H that is strongly 2-universal.