1) (10 pts) ALG (Analysis and Critical Thinking: AVL Trees, Hash Tables, and Heaps)

a) (1 pt) Using big-oh notation, what is the **best-case** runtime for inserting an integer into an AVL tree that contains *n* integers?

O(log n) Grading: All or nothing.

b) (1 pt) Using big-oh notation, what is the **worst-case** runtime for inserting an integer into an AVL tree that contains *n* integers?

O(log n) Grading: All or nothing.

c) (2 pts) What is the worst-case runtime for insertion into a hash table with *n* elements, assuming we use quadratic probing to resolve collisions? (You may assume that our hash table satisfies all conditions necessary to ensure that quadratic probing won't get stuck in an infinite loop.)

O(n) Grading: All or nothing.

d) (2 pts) Given the following hash table, suppose we know that no strings have been deleted, but we don't know the order in which these three strings were inserted into the hash table. If we used linear probing to resolve collisions, what are all the possible hash values for the string "of" (assuming those hash values are modded by the table size)?

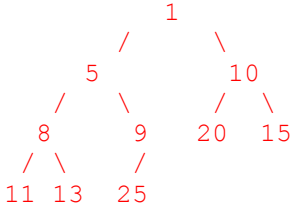
		pied	piper	of		Hamelin
0	1	2	3	4	5	6

Possible hash values: 2, 3, or 4
Grading: Give one point for each correct index given, and subtract half a point for each incorrect index given. Then take the floor.

e) (2 pts) Using big-oh notation, what is the **worst-case** runtime for deletion from a minheap that contains *n* elements?

O(log n) Grading: All or nothing.

f) (2 pts) Draw a minheap that contains 10 elements and which will incur the worst-case runtime if we call deleteMin() on it.



Note: Actual values may vary from answer to answer, but it's important that the percolate down operation takes the value to the left of the root, then left again, and then it can go left or right (doesn't matter).

Grading: 1 pt for a valid minheap, 1 pt for a minheap where the node in 25's position ends up where 11 or 13 are.