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?

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?

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.)

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, so the only valid values are 0 through 6)?

| | | | | | | |
|---|---|---|---|---|---|---|
| | | pied | piper | of | | Hamelin |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

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

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