Grant Nakashima

CS 261-400

Assignment 3 Part 2

Linked List		
N	Memory Used (KB)	Time (ms)
1000	0	0
2000	0	0
4000	0	30
8000	0	130
16000	0	510
32000	496	2060
64000	1496	8060
128000	3492	32360
265000	7496	235210

Dynamic Memory		
N	Memory Used (KB)	Time (ms)
1000	0	0
2000	0	0
4000	0	20
8000	0	100
16000	0	380
32000	0	1520
64000	0	6090
128000	140	24350
265000	644	97480

Which of the implementations uses more memory? Explain why.

- The linked list uses more memory. Lists have to hold more memory for the next and previous nodes. Arrays don't need to allocate memory for that additional information so it uses less memory than a linked list. Array sets aside a set amount of memory when it resizes. A linked list grows organically and will allocate memory depending on how much it will need.

Which of the implementations is the fastest? Explain why.

- Dynamic arrays are faster for the contains function. A linked list has to traverse the list from the beginning to get to a specific element. A dynamic array allows direct reference to every element so you are able to access them in constant time.

Would you expect anything to change if the loop performed remove() instead of contains()? If so,what?

- Yes. In the dynamic array, you will have to shift every element after the remove function has been called. This will be like resizing so it will end up being O(n). The total time for an array is access time + O(n). A linked list needs to find the element and then delete it. It doesn't need to shift all of the elements like an array. The total time for a linked list is access time + O(1).