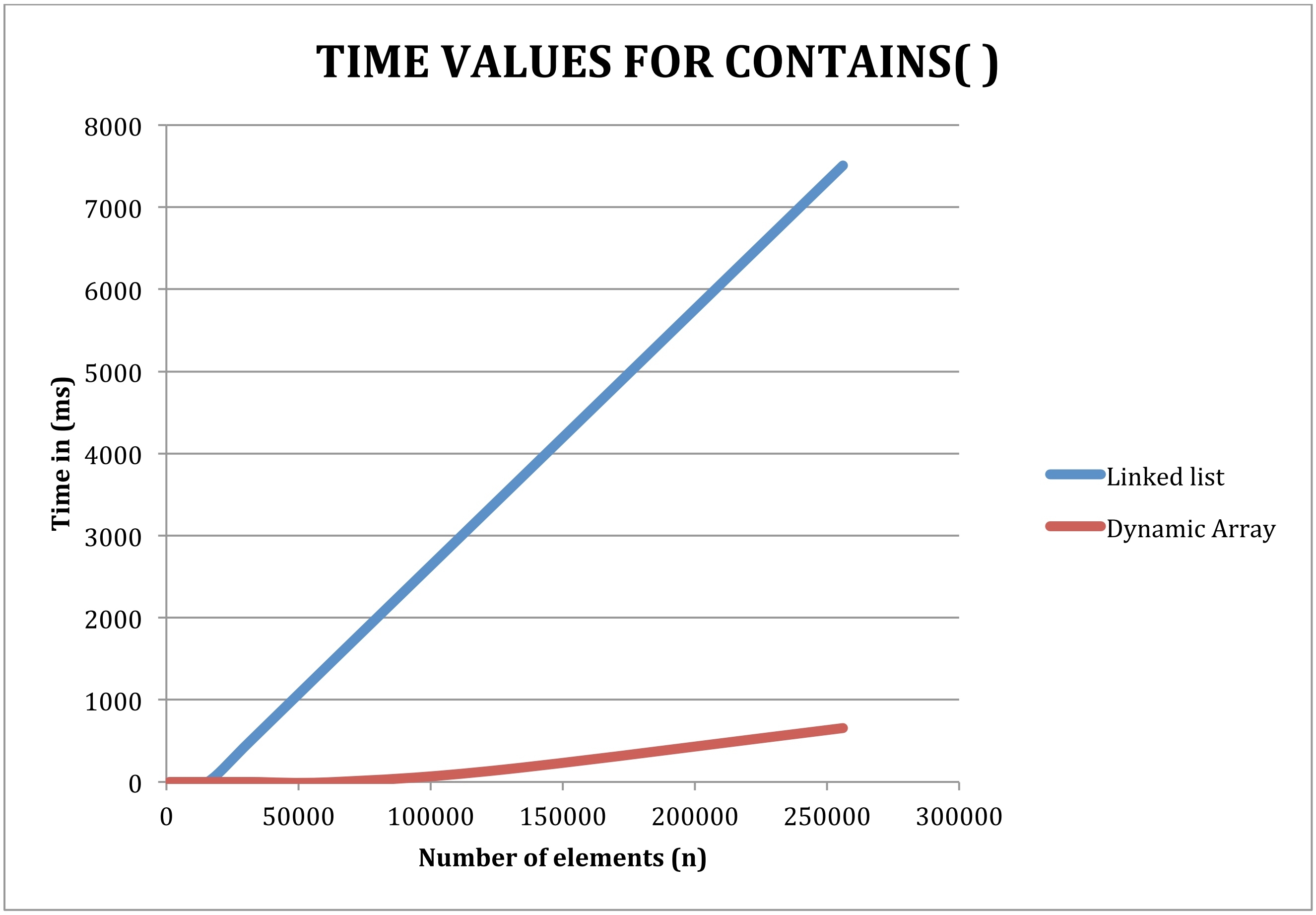
Name: Carlos Carrillo
Date: 01/28/2016
CS 261
Problem 2: Comparison

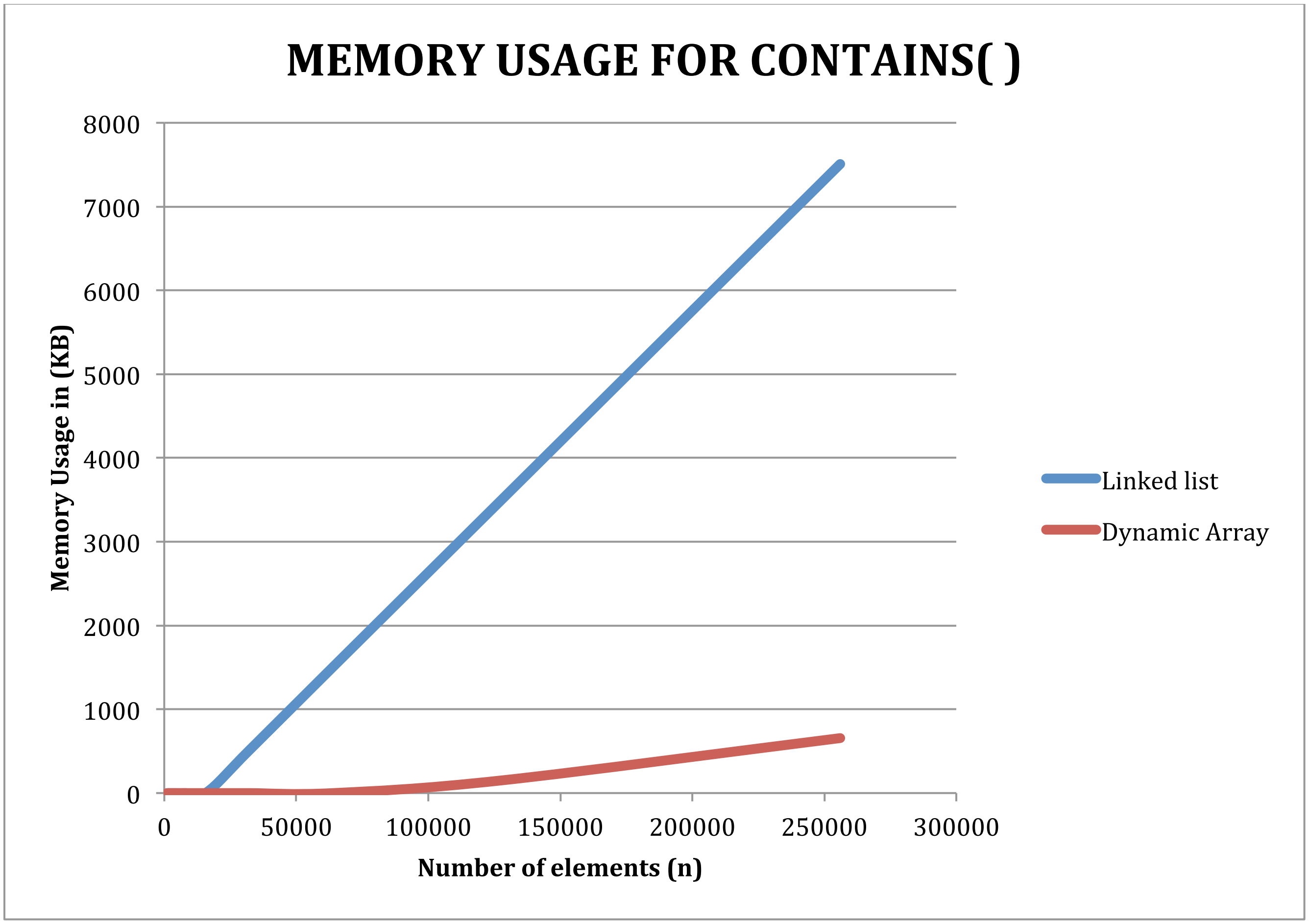
TIME VALUES FOR CONTAINS (Linked list Vs. Dynamic Array)

Number of elements (n)	Linked list	Dynamic Array
1000	0 ms	0 ms
2000	0 ms	0 ms
4000	30 ms	20 ms
8000	130 ms	100 ms
16000	510 ms	390 ms
32000	2010 ms	1530 ms
64000	8260 ms	6100 ms
128000	33610 ms	24440 ms
256000	223960 ms	97570 ms



MEMORY USAGE FOR CONTAINS (Linked list Vs. Dynamic Array)

Number of elements (n)	Linked list	Dynamic Array
1000	0 KB	0 KB
2000	0 KB	0 KB
4000	0 KB	0 KB
8000	0 KB	0 KB
16000	8 KB	0 KB
32000	504 KB	0 KB
64000	1508 KB	0 KB
128000	3508 KB	152 KB
256000	7508 KB	656 KB



1. Which of the implementations uses more memory? Explain why.

   According to the data collected, in this case, the Linked-list implementation uses more memory. This is because the Linked-List implementations need memory to store not only the value, but also the next and previous pointer, whereas the Dynamic Array only needs to store the value.

2. Which of the implementations is the fastest? Explain why.

   According to the data collected, in this case, the dynamic array implementation is faster. Although Linked-list performance is usually faster asymptotically than dynamic arrays for insertion and removal operations, dynamic array implementations will typically be faster for traversal operations like contains. This is in part because an array data is tightly packed into consecutive memory, whereas list nodes are allocated individually and potentially scattered throughout memory. So a large chunk of the array can be in memory cache at once during traversal, but each link in the list may need to be retrieved individually into the cache causing a lot of cache misses during traversal.

   On the other hand, the contains () method for both implementations needs to iterate from the end of the list and make comparisons with each element until it finds what it needs. In both cases that would be an O(n) operation. So, for lower values linked list may be faster but for higher values dynamic array should dominate.

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

   I think the Linked-list implementation would be a little bit faster than the Dynamic Array when running the remove( ) function. This is because the Dynamic Array must move all the elements up from the index where the wanted element is found. This can obviously slow down its performance drastically to O(n) in the worst case scenario. On the contrary, the Linked-List implementation only needs to remove the links and deallocate the memory, which can be done in O(1).

   However, this O(1) run time added to the already existing O(n) will clearly become a O(n), which would be very similar to the running time generated by the Dynamic array implementation, so the difference would be almost imperceptible. The only way to really know this difference would be by carefully measure the two processes. But in theory, the running time of the Linked-list implementation should be slightly smaller/faster.