Name: Kang-Hua Wu
SJSU ID: 011470368

Assignment 9B

1. Explain the difference in timings between the vector and the linked list for the prepend operation.

Vector has much larger operating time than list does. For vectors, prepend operation means shifting all elements right by one index. For lists, prepend operation simply creates a new link in front of the old list.

2. Explain the difference in timings between the vector and the linked list for the get value operation.

Vector has less operating time than linked list. Vector has a predefined space in memory for each index upon declaration. Therefore, finding for specific index is very fast — just offset by index amount from starting pointer. In contrast, for linked list, we have iterate to the given index for its value.

3. Explain the difference in timings between the vector and the linked list for the search operation.

Both have comparable operating time since we have to iterate vector and linked list before finding the the right value. Linked lists have slight higher operating time because they require more get time than vectors do (as discussed in question 2).

4. Explain the difference in timings between the vector and the linked list for the remove operation.

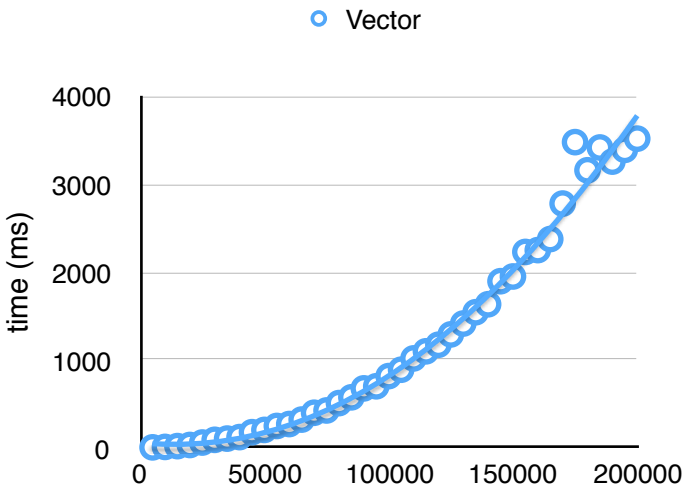
Answer for this question is very similar to prepend operation. Every time the vector removes its first element, it has to shift the new vector one index left but linked list only has to delete the first node and make the second element as the new starting node.

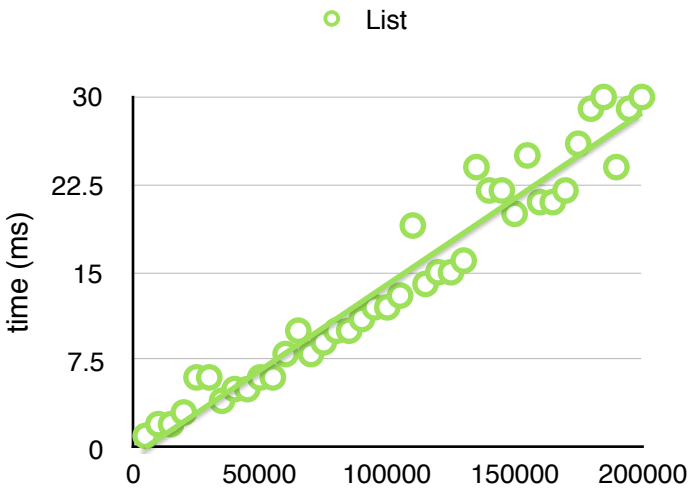
5. Explain the difference in timings between the vector and the linked list for the append operation.

Both vector and linked list have a very operating time. Vector shows only slightly faster due to slight complex manipulation of pointer while appending a new node to a linked list.

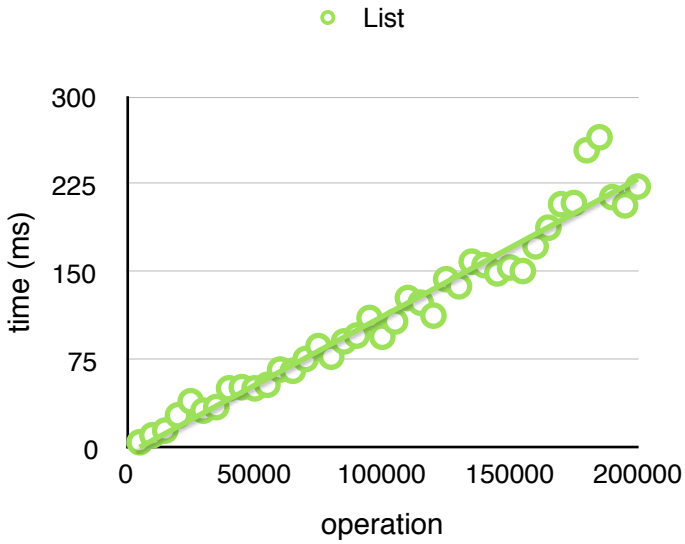
6. Identify and graph the rates of growth (linear, quadratic, exponential, logarithmic, etc.) of the prepend operation as the size increases for both the vector and the linked list, and explain the growth rates.

Vector has quadratic growth rate: for each insertion in the beginning of vector, vector has to shift right one index for every element and should have linear growth rate depending on the size of vector. In addition, considering prepend here is inserting element while growing its elements from 0 to n, vector is shifting in quadratic rate.
Linked list should categorize as constant growth rate although behave as linearity in small scale. Elapsed time barely increase from 5 ms handling 5000 nodes of linked list to 30 ms handling 200000 nodes.





7. Identify, graph, and explain the rate of growth of the get value operation for the linked list.
Linked list can not directly access to specific index position. It can only access by iterating from the starting node or the ending node. Therefore, the growth rate increase with increasing length of linked list.



8. Identify, graph, and explain the rates of growth of the search operation for both the vector and the linked list.

Both vector and linked list have linear growth rate. From the charts below, we can see that the slope for linked list is slightly higher than vector's slope because vector has a slight faster than linked list does when accessing its value for a given index.