13. Suppose that when a node is added to a singly linked list, it becomes the new last node. Assuming the list header is named h, and the link field is named next, give the pseudocode algorithm for this approach to the Insert algorithm.

Node n = new Node();

compute final; // final is the address of the last node in the list.

This computation requires stepping down the list and following the pointers. There is a memory access at every step.

final.next = n;

n.data = input.deepCopy();

14. Give the dominant term in the speed function of the Insert algorithm described in the previous exercise.

O(n+3) -> O(n)

15. The reference variable p references a node in the middle of a long singly linked list, and q points to the node just before the node that p references.

a) Give the standard graphical representation of the list including the reference variables p and q.

15a and 15b are on the accompanying JPEG.

b) Modify the graphic to show the deletion of both the node p points to and the node that follows it.

c) Give the pseudocode to accomplish the deletion of the two nodes.

Advance to p and q by searching list.

Node final = p.next.next; // Header for bottom part of list.

q.next = final; //cut off the nodes we don't want.

(The last two steps could be combined: q.next = p.next.next)

20. Give the ratio of the average speed of an Unsorted-Optimized array structure to the average speed of a Singly Linked List structure, assuming each structure contains one million information nodes and all operations are equally probable. Note: See **Table 4.1** in the text for the average speed equations.

With n = 106 we only need the limit of large n.

ratio = (3n+6)/(4.5n+12) -> 3/4.5 = 0.67

21. Describe the garbage collection method for the SinglyLinkedList structure, and give the line number of the code presented in **Figure 4.15** of the text that accomplishes (i.e., actually initiates) the “garbage collection.”

*Garbage collection in the linked list is something done automatically by the memory manager. It is initiated by line 37 which moves the pointer past the deleted node. At that point the deleted node has no pointer so the memory manager takes over.*

22. A Singly Linked List structure is used to store a data set. Calculate its density if:

a) Each of the client's information nodes contains 8 bytes of information, and there are 50 nodes in the data set.

w=8, n=50. D = 0.49

b) Each of the client's information nodes contain 200 bytes of information, and there are one million nodes in the data set.

w=200, n=106. D = 0.96

23. Give a plot showing the variation in density with the number of nodes*, n*, stored in a SinglyLinkedList structure. Assume each node contains 10 information bytes and the range of *n* is 2   
  
**Note:** This question expects you to do a graph. You can do this by using Microsoft Excel with n in one column and the density in another column. Calculate the density using the formula D = 1 / (1 + 12/ (n\* w) + 8 / w), where w =10. Points for n should include (2, 10, 20, 30, 40, …, 90, 100). Use the X Y Scatter chart to create your graph. You then can copy and paste your resulting graph into this Word document.

Density vs. n

24. Give an example of when it would be more efficient to use an iterator to access the nodes in a singly linked list.

*The example in the book is to update part of an entry. The normal update method for a linked list would not work as quickly. The advantage of the iterator is it visits every node exactly once. The usual search method requires starting at the beginning every time.*