

EX 13.1 (Explain what will happen if the steps depicted in Figure 13.4 are reversed.)

If the steps in Figure 13.4 were reversed the linked list would become unnavigable. The reference to the front of the list being pointed to the newly added node before the current first node has a pointer then you would create a hole in the chain of nexts making a node *uncallable*.

EX 13.2 (Explain what will happen if the steps depicted in Figure 13.5 are reversed.)

If you delete the first node in a list before reset the reference to the front of the list then it will not point to the current second node in the list and you will lose access to the top of the list. You will lose the list.

EX 13.4

Book example of push:

"To position the newly created node at the top of the stack, we must set its next reference to the current top of the stack and reset the top reference to point to the new node. We must also increment the count variable."

```
public void push(T element)
{
    LinearNode<T> temp = new LinearNode<T>(element);

    temp.setNext(top);
    top = temp;
    count++;
}
```

My attempt at add:

```
public void add(T element)
{
    LinearNode<T> temp = new LinearNode<T>(element);

    LinearNode<T> cur = top;

    while (cur.getNext() != null) // does until the current top is null
(end)
{
    cur = cur.getNext();
}
cur.setNext(temp); // sets current element to end
```

```
    count++;
}
```

Time Complexity: (O)n Linear time complexity.

EX 13.5

Rear reference:

```
private LinearNode<T> rear; // hypothetical rear pointer

public void add(T element)
{
    LinearNode<T> temp = new LinearNode<T>(element);

    rear.setNext(temp);
    rear = temp;
    count++;
}
```

Time Complexity: (O)1 Constant time complexity. Solely three operations thanks to rear reference.

EX 13.6

The count++ operations serves as the length of the linked list, why you see it decrease in the pop function. This is similar to the rear reference, as both as time complexity reducing measures. Without the count++ operation the linked list would have to be counted every single time the length was needed for an operation.