10.2-7.

Give a Θ (n)-time nonrecursive procedure that reverses a singly linked list of n elements. The procedure should use no more thatn sonstant storage beyond that needed for the list itself.

Answer.

The List-Reverse procedure reverse a given list while walking down to its tail. We employ three auxiliary pointers so as to capture both the predecessor and successor of the currently visited object.

```
List-Reverse($L$)
if $L.head$ == NIL
    error "empty list"
elseif $L.head.next$ == NIL
    done
else
    $p_1 = L.head.next$
    $L.head.next =$ NIL
    $p_2 = p_1.next$
    $p_1.next = L.head$
    while $p_2 \neq$ NIL
        $p_3 = p_2.next$
        $p_2.next = p_1$
        $p_1 = p_2$
        $p_2 = p_3$
    $L.head = p_1$
```

Figure 1 gives a rather intuitive illustration of it while reversing a sample list. The List-Reverse procedure takes Θ (n) time to reverse a list of n elements.

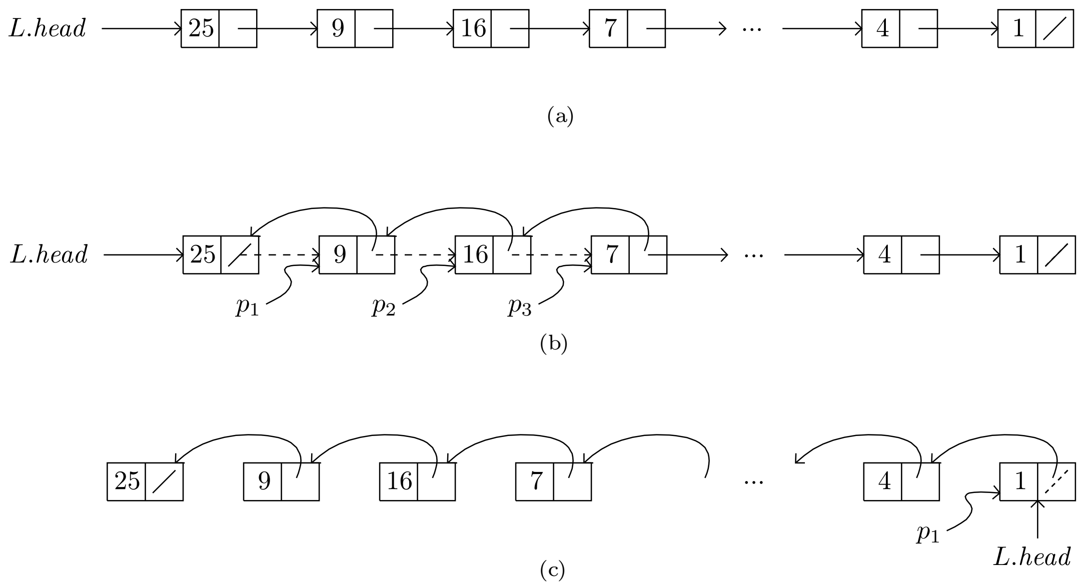

Figure 1. (a) a singly linked list L representing the dynamic set $\{25, 9, 16, 7, ..., 4, 1\}$. Each element in the list is an object with attributes for the key and pointer to the next object. **(b)** The list structure after the first two iterations of reversion, the dashed arrows indicates the destroyed pointers. **(c)** The result of the call to List-Reverse(L), where $L.head$ points to the object with key 1.

Email address: informlarry@gmail.com