Exercise Session 06

Exercise 1.
In class we have seen that using arrays for implementing stack may lead to underflow and overflow problems. Propose an alternative implementation of the stack operations that use doubly linked lists. What is the worst-case running time for the POP and PUSH operations? May we still incur in situations where the stack underflows or overflows?

Exercise 2.
Exercises 10.4-2 and 10.4-3 CLRS pp. 248.

Exercise 3.
Singly linked lists are a variant of linked lists where each element x has two attributes, $x.key$ that stores the key, and $x.next$ that points to the next element in the list. Can you implement the dynamic-set operation INSERT on a singly linked list in $O(1)$ time? How about DELETE?

Exercise 4.
Give a $\Theta(n)$-time nonrecursive procedure that reverses a *singly* linked list of n elements. The procedure should use no more than constant storage beyond that needed for the list itself.

Exercise 5.
Implement a queue by a singly linked list Q. The operations ENQUEUE and DEQUEUE should still take $O(1)$ time.