***Level1: B. Describe, in pseudo-code, how to implement the stack ADT using a DLinkedList. What is the running time of the push() and pop() methods in this case?***

The top element of stack ADT is store at the first node of the DLinkedList.

The space used is O(n) and each operations of the stack ADT take O(1) time.

The running time of the push() and pop() is O(1).

***Level1: Describe, in pseudo-code, how to implement the queue ADT using a DLinkedList. What is the running time of the enqueue() and dequeue() methods in this case?***

The front element of queue ADT is stored at the first node

The rear element of queue ADT is stored at the last node.

The space used is O(n) and each operations of the queue ADT take O(1) time.

The running time of the enqueue() and dequeue() is O(1).

***Level2: C-2.2 Describe, in pseudo-code, how to implement the queue ADT using two stacks. What is the running time of the enqueue() and dequeue() methods in this case?***

Implement the queue ADT using two stacks

1. stack1 =[], stack 2 =[]
2. Implement enqueue() by pushing element into the stack1  
   enqueue(element) { stack1.push(element);}
3. Implement dequeue() by popping elements out of stack1 and pushing them into stack2, which reverse the order and then pop the element out of stack2

dequeue() {

If stack1.isEmpty() then throw EmptyQueueException

While !stack1.isEmpty do{

element = stack1.pop;

stack2.push(element);}

return stack2.pop();

}  
  
The running time of the enqueue() is O(1) and the running time of dequeue is O(n) because we have to transfer all elements from stack1 to stack2.