Problem 2:

1. The time complexity of remove ( ) fucntion is O(1) because there is always a pointer pointing to the tail of the function so if we want to remove the least recently added node then we have to point another pointer to the tail. Since it always removes from the tail the size of the linked list doesn’t matter.

**public** **int** remove() {

Node last = tail;

tail.next = last;

last.next = **null**;

**return** last.data;

}

1. The implementation of the add ( ) function is O(n) time complexity because as the input increases the time also increases linearly since the input is added to the head and the head has to readjust itself to the front every time.

**public** **void** add(**int** a){

Node first = head;

head = **new** Node ();

head.data = a;

head.next = first;

size++;

}