::::::::::::::
Lab4Status.txt
::::::::::::::
Problem 1: compiles, runs correctly on all provided input
Problem 2: Completed::::::::::::::
Lab4Conclusions.txt
::::::::::::::
This lab was easily the most helpful lab as far as learning is concerned. I found the implementation of a Circular Doubly Linked Structure to be insanely efficient and is easily the most advanced material I have dabbled with thus far as a programmer. While I won't say I'm an expert in calculating space/time complexity **for** algorithms, I still found the investigation to be informative and hopefully in the future I can utilize it to figure out what methods to implement **for** programs in the future. ::::::::::::::
DNode.java
::::::::::::::

```
/**
 * Purpose: Data Structure and Algorithms Lab 4
 * Status: Complete and thoroughly tested
 * Last update: 02/13/23
 * Submitted:  02/13/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.02.13
 */

public class DNode
{
    private Object item;
    private DNode next; // next item
    private DNode back; // previous item

    public DNode(Object newItem)
    {
        item = newItem;
        next = this;
        back = this;
    } // end constructor

    public DNode(Object newItem, DNode nextNode, DNode lastNode)
    {
        item = newItem;
        next = nextNode;
        back = lastNode;
    } // end constructor

    public void setItem(Object newItem)
    {
        item = newItem;
    } // end setItem

    public Object getItem()
    {
        return item;
    } // end getItem

    public void setNext(DNode nextNode)
    {
        next = nextNode;
    } // end setNext

    public DNode getNext()
    {
        return next;
    } // end getNext

    public void setBack(DNode lastNode)
    {
        back = lastNode;
        // end setLast
    }

    public DNode getBack()
    {
        return back;
        // end getLast
    }
} // end class DNode::::::::::::::
```

ListCDLSBased.java
::::::::::::::

```
/**
 * Purpose: Data Structure and Algorithms Lab 4
 * Status: Complete and thoroughly tested
 * Last update: 02/13/23
 * Submitted: 02/13/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 *
 * @author: Antonio Rosado
 * @version: 2023.02.13
 */

public class ListCDLSBased implements ListInterface
{
    private DNode head; // 'beginning' of list
    private int numItems; // number of items in list

    public ListCDLSBased()
    {
        head = null;
        numItems = 0;
    }

    /**
     * Check if DNode is empty
     *
     * @return head == null
     */
    public boolean isEmpty()
    {
        return head == null;
    }

    /**
     * Return size of DNode
     *
     * @return size of DNode
     */
    public int size()
    {
        return numItems;
    } // end size
```

```
    /**
     * Get item from DNode
     *
     * @param int index index to find
     * @return DNode curr item found
     */
    private DNode find(int index) throws ListIndexOutOfBoundsException {
        DNode curr = head;
        int num = 0;
        if (index > (size() / 2))
        {
            while (num < (size() - index))
            {
                curr = curr.getBack();
                num++;
            }
        }

        else
        {
            while (index > num)
            {
                curr = curr.getNext();
                num++;
            }
        }
        return curr;
    }

    /**
     * Get item from DNode
     * @param int index index of item
     * @return Object nodeItem item from specified index
     */
    public Object get(int index) throws ListIndexOutOfBoundsException
    {
        if (index >= 0 && index < size())
        {
            // get reference to node, then data in node
            DNode curr = find(index);
            Object nodeItem = curr.getItem();
            return nodeItem;
        }

        else
        {
            throw new ListIndexOutOfBoundsException("List index out of bounds exce
ption on get");
        } // end if
        // end get
    }

    /**
     * Add item to DNode
     * @param int    index index of item
     * @param Object item item Object
     */
    public void add(int index, Object item) throws ListIndexOutOfBoundsException
    {
        if (size() == 0)
        {
            DNode temp = new DNode(item); // new DNode that references itself
            head = temp;
            numItems++;
            return;
        }

        else if (index >= 0 && index <= size())
        {
            if (index == 0)
            {
                DNode newNode = new DNode(item, head, head.getBack());
                head.getBack().setNext(newNode);
                head.setBack(newNode);
                head = (index == 0) ? newNode : head;
            }

            else
            {
                // store previous DNode which will then have its following index r
eference new
                // DNode
                DNode prev = find(index - 1);
                DNode newNode = new DNode(item, prev.getNext(), prev);
                prev.getNext().setBack(newNode);
                prev.setNext(newNode);
            }
            numItems++; // update num of items
        }

        else
        {
            throw new ListIndexOutOfBoundsException("List index out of bounds on a
dd.");
        }
    }

    /**
     * Remove item from DNode
     * @param int index index of item
     */
    public void remove(int index) throws ListIndexOutOfBoundsException
    {
        if (index >= 0 && index < size())
        {
            if (index == 0) // if item is first or 0
            {
                // delete the first DNode from the list
                DNode prev = head.getBack(); // retrieve previous item
                head = head.getNext(); // set head to second or index 1
                head.setBack(prev); // set previous to new head
            }

            else
            {
                // delete the DNode after the previous DNode
                // references, save reference to DNode
                DNode prev = find(index - 1); // find previous index
                DNode curr = prev.getNext(); // retrieve current or next index
                prev.setNext(curr.getNext());
                prev.setBack(prev.getBack());
            } // end if
            numItems--; // decrement num of items
        } // end if
```

```
        else
        {
            throw new ListIndexOutOfBoundsException("List index out of bounds exce
ption on remove");
        } // end if
    } // end remove

    /**
     * Remove all items from DNode
     */
    public void removeAll()
    {
        numItems = 0; // if empty, no items
        head = null; // no head if empty
    }

    /**
     * Returns a string value of DNode items
     */
    public String toString()
    {
        String list = "";
        DNode curr = head;
        for (int index = 0; index <= numItems - 1; index++)
        {
            list += curr.getItem() + " "; // retrieve and collect item
            curr = curr.getNext(); // retrieve next item
        }
        return list.toString(); // collection becomes a string
    }


    /**
     * Returns a string value of DNode items reversed
     */
    public String toStringR()
    {
        String reversed = "";
        DNode curr = head;
        for (int index = 0; index <= numItems - 1; index++)
        {
            reversed += curr.getBack().getItem() + " "; // retrieve and collect it
em
            curr = curr.getBack(); // retrieve previous item
        }
        return reversed.toString();
    }
}
::::::::::::::
Lab4P1Driver.java
::::::::::::::
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Lab4P1Driver
{
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));
    public static void main (String[] args) throws IOException
    {
        ListCDLSBased myList = new ListCDLSBased();
        Lab4P1Driver driver = new Lab4P1Driver();
        boolean exit = false;
        int pos = -1;
        while (!exit) {
            System.out.println("Select from the following menu: \n"
                               + "\t 0. Exit the program \n"
                               + "\t 1. Insert item into the list \n"
                               + "\t 2. Remove item from the list \n"
                               + "\t 3. Get item from the list \n"
                               + "\t 4. Clear the list \n"
                               + "\t 5. Display size and content of the list in or
der and in reversed order \n"
                               + "\t 6. Delete the smallest and largest item in th
e list \n"
                               + "\t 7. Reverse the list \n");

            System.out.print("Make your menu selection now: ");
            int input = Integer.parseInt(stdin.readLine().trim());
            System.out.println(input);
            // possible cases for initial input
            switch (input) {
            case 0:
                System.out.println("Exiting program... good bye");
                exit = true;
                break;

            case 1:
                try
                {
                    System.out.println("You are now inserting an item into the lis
t.");
                    System.out.print("\t Enter item: ");
                    Object item = stdin.readLine().trim();
                    System.out.println(item);

                    System.out.print("\t Enter the position to insert item in: ");
                    pos = Integer.parseInt(stdin.readLine().trim());
                    System.out.println(pos);
                    if (pos <= myList.size())
                    {
                        myList.add(pos, item);
                        System.out.println("Item " + item + " inserted in position
 " + pos + " in the list.");
                    }
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("Position specified is out of range!");
                }
                break;

            case 2:
                try
                {
                    System.out.println("You are now removing an item from the list
.");
                    System.out.print("\t Enter position to remove item from: ");
                    pos = Integer.parseInt(stdin.readLine().trim());
                    System.out.println(pos);
                    System.out.println("Item " + myList.get(pos) + " removed from
position " + pos + " in the list.");
                    myList.remove(pos);
```

```
                }
                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("Position specified is out of range!");
                }
                break;

            case 3:
                try
                {
                    System.out.print("\t Enter position to retrieve item from: ");
                    pos = Integer.parseInt(stdin.readLine().trim());
                    System.out.println(pos);
                    System.out.println("Item " + myList.get(pos) + " retrieved fro
m position " + pos + " in the list.");
                }

                catch(ListIndexOutOfBoundsException e)
                {
                    System.out.println("Position specified is out of range!");
                }
                break;

            case 4:
                if(myList.isEmpty())
                {
                    System.out.println("List is empty, nothing to clear!");
                }

                else
                {
                    System.out.println("Clearing list...");
                    myList.removeAll();
                    System.out.println("List cleared.");
                }
                break;

            case 5:
                if(!(myList.size() == 0))
                {
                    System.out.println("In order: List of size " + myList.size() +
 " has the following items: " + myList.toString());
                    System.out.println("In reverse order: List of size " + myList.
size() + " has the following items: " + myList.toStringR());
                }

                else
                {
                    System.out.println("List is empty.");
                }
                break;

            case 6:
                driver.displayAndDeleteLargeAndSmall(myList);
                break;

            case 7:
                if(!(myList.isEmpty()))
                {
                    System.out.println("List has been reversed.");
                    ListCDLSBased reversed = driver.reverse(myList);
                    System.out.println("\t Here is the content: " + reversed.toStr
ing());
                }

                else
                {
                    System.out.println("List is empty... nothing to reverse!");
                }
                break;
            }
        }
    }

    /**
     * Find largest and smallest items from DNode collection lexicographically
     * @param ListCDLSBased myList              list to be iterated
     * @param int[] numbers                 Array of largest and smallest values
     */
    public void findIndexLargeAndSmall(ListCDLSBased myList, int[] numbers)
    {
        int size = myList.size();
        int smallIndex = 0;
        int largeIndex = 0;
        String curr = "";
        String smallestValue = myList.get(0).toString();
        String largestValue = myList.get(0).toString();
        for (int index = 0; index < size; index++)
        {
            curr = myList.get(index).toString();
            if(curr.compareTo(smallestValue) <= 0)
            {
                smallIndex = index;
                smallestValue = myList.get(smallIndex).toString();
            }

            else if(curr.compareTo(largestValue) >= 0)
            {
                largeIndex = index;
                largestValue = myList.get(largeIndex).toString();
            }
        }
        numbers[0] = smallIndex;
        numbers[1] = largeIndex;
    }

    /**
    * Display and delete largest and smallest items from DNode collection lexicogr
aphically
    * @param ListCDLSBased myList        list to be iterated
    */
    public void displayAndDeleteLargeAndSmall(ListCDLSBased myList) throws ListInd
exOutOfBoundsException
    {
        if(myList.size() == 0)
        {
            System.out.println("List is empty, nothing to delete!");
        }
        else if(myList.size() == 1)
        {
            System.out.println(myList.toString() + " is deleted.");
            myList.remove(0);
        }
```

```
        else
        {
            int numbers[] = new int[2];
            findIndexLargeAndSmall(myList, numbers);
            System.out.println("Smallest item " + myList.get(numbers[0]) + " delet
ed.");
            System.out.println("Largest item " + myList.get(numbers[1]) + " delete
d.");
            myList.remove(numbers[0]);
            myList.remove(numbers[1]);
        }
    }

    /**
    * Reverse collection and return it reversed
    * @param ListCDLSBased myList        list to be iterated
    * @return temp                       reversed list
    */
    public ListCDLSBased reverse(ListCDLSBased myList)
    {
        ListCDLSBased temp = new ListCDLSBased();
        int size = myList.size();
        for (int index = 0; index < size; index++)
        {
            temp.add(index, myList.get(size - 1 - index));
        }
        myList = temp;
        return myList;
    }
}
```

```
::::::::::::::
Lab4P1Sampleruns.txt
::::::::::::::
Select from the following menu:
         0. Exit the program
         1. Insert item into the list
         2. Remove item from the list
         3. Get item from the list
         4. Clear the list
         5. Display size and content of the list in order and in reversed order
         6. Delete the smallest and largest item in the list
         7. Reverse the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
         0. Exit the program
         1. Insert item into the list
         2. Remove item from the list
         3. Get item from the list
         4. Clear the list
         5. Display size and content of the list in order and in reversed order
         6. Delete the smallest and largest item in the list
         7. Reverse the list

Make your menu selection now: 6
List is empty, nothing to delete!
Select from the following menu:
         0. Exit the program
         1. Insert item into the list
         2. Remove item from the list
         3. Get item from the list
         4. Clear the list
         5. Display size and content of the list in order and in reversed order
         6. Delete the smallest and largest item in the list
         7. Reverse the list

Make your menu selection now: 7
List is empty... nothing to reverse!
Select from the following menu:
         0. Exit the program
         1. Insert item into the list
         2. Remove item from the list
         3. Get item from the list
         4. Clear the list
         5. Display size and content of the list in order and in reversed order
         6. Delete the smallest and largest item in the list
         7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
         Enter item: Data
         Enter the position to insert item in: 0
Item Data inserted in position 0 in the list.
Select from the following menu:
         0. Exit the program
         1. Insert item into the list
         2. Remove item from the list
         3. Get item from the list
         4. Clear the list
         5. Display size and content of the list in order and in reversed order
         6. Delete the smallest and largest item in the list
         7. Reverse the list

Make your menu selection now: 5
In order: List of size 1 has the following items: Data
In reverse order: List of size 1 has the following items: Data
Select from the following menu:
         0. Exit the program
         1. Insert item into the list
         2. Remove item from the list
         3. Get item from the list
         4. Clear the list
         5. Display size and content of the list in order and in reversed order
         6. Delete the smallest and largest item in the list
         7. Reverse the list

Make your menu selection now: 7
List has been reversed.
         Here is the content: Data
Select from the following menu:
         0. Exit the program
         1. Insert item into the list
         2. Remove item from the list
         3. Get item from the list
         4. Clear the list
         5. Display size and content of the list in order and in reversed order
         6. Delete the smallest and largest item in the list
         7. Reverse the list
```

```
Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Beverly
        Enter the position to insert item in: 0
Item Beverly inserted in position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Beverly Data
In reverse order: List of size 2 has the following items: Data Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Jean-Luc
        Enter the position to insert item in: 5
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Beverly Data
In reverse order: List of size 2 has the following items: Data Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Jean-Luc
        Enter the position to insert item in: 2
Item Jean-Luc inserted in position 2 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Geordi
        Enter the position to insert item in: 2
Item Geordi inserted in position 2 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Worf
        Enter the position to insert item in: 3
Item Worf inserted in position 3 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 5 has the following items: Beverly Data Geordi Worf Jean-Lu
c
In reverse order: List of size 5 has the following items: Jean-Luc Worf Geordi Dat
a Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Jean-Luc Worf Geordi Data Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
```

```
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Jean-Luc Worf Geordi Data Beverly
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 6
Smallest item Beverly deleted.
Largest item Worf deleted.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 3 has the following items: Data Geordi Worf
In reverse order: List of size 3 has the following items: Jean-Luc Worf Geordi
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Jean-Luc Geordi Data
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Jean-Luc Geordi Data
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 2
You are now removing an item from the list.
        Enter position to remove item from: 9
Position specified is out of range!
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 2
You are now removing an item from the list.
        Enter position to remove item from: 2
Item Jean-Luc removed from position 2 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Data Geordi
In reverse order: List of size 2 has the following items: Jean-Luc Worf
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 2
You are now removing an item from the list.
        Enter position to remove item from: 0
Item Data removed from position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list
```

```
Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Will
        Enter the position to insert item in: 0
Item Will inserted in position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Will Geordi
In reverse order: List of size 2 has the following items: Jean-Luc Worf
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 3
        Enter position to retrieve item from: 1
Item Geordi retrieved from position 1 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 3
        Enter position to retrieve item from: 0
Item Will retrieved from position 0 in the list.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 3
        Enter position to retrieve item from: 8
Position specified is out of range!
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 2 has the following items: Will Geordi
In reverse order: List of size 2 has the following items: Jean-Luc Worf
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 6
Smallest item Geordi deleted.
Largest item Will deleted.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content:
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 1
You are now inserting an item into the list.
        Enter item: Velcro
        Enter the position to insert item in: 0
Item Velcro inserted in position 0 in the list.
Select from the following menu:
```

```
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
In order: List of size 1 has the following items: Velcro
In reverse order: List of size 1 has the following items: Velcro
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 7
List has been reversed.
        Here is the content: Velcro
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 6
Velcro  is deleted.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 5
List is empty.
Select from the following menu:
        0. Exit the program
        1. Insert item into the list
        2. Remove item from the list
        3. Get item from the list
        4. Clear the list
        5. Display size and content of the list in order and in reversed order
        6. Delete the smallest and largest item in the list
        7. Reverse the list

Make your menu selection now: 0
Exiting program... good bye
::::::::::::::
```

Lab4P2.txt
::::::::::::::
a) Space Complexity analysis:

The CDLS implementation uses **class** DNode that itself contains 3 instance variables
.
Those being:
An object reference to the item stored in DNode.
A reference to the next DNode in the list.
A reference to the previous DNode in the list.

ListCDLSBased contains two instance variables.
Those being:
An Object reference to the first item in DNode
An **int** to keep track of the number of items in DNode

Each object reference should take 4 bytes so each DNode would require 4 * 4 = 16 b
ytes. The number of DNodes in the list is stored in the **"numItems"** variable which
takes 4 bytes. So, the total space complexity of the CDLS implementation is O(16 *
 n + 4) = O(16n + 4), where n is the number of items in the list.

b) Time Complexity analysis:

get(**int** index) - The **"find"** method is called to find the item at the specified ind
ex. The find method has a time complexity of O(n/2) in the worst-**case** scenario whe
re n is the number of items in the list. So, the overall time complexity of the **"g
et"** method is O(n/2) + O(1) = O(n/2), where O(1) represents the time taken to **retu
rn** the node item.

add(**int** index, Object item): The **"find"** method is again called to find the DNode b
efore the specified index. The find method has a time complexity of O(n/2) in the
worst-**case** scenario where n is the number of items in the list. So, the overall ti
me complexity of the **"add"** method is O(n/2) + O(1) = O(n/2), where O(1) represents
 the time taken to add the item.

remove(**int** index): The **"find"** method is called to find the DNode at the specified
index. The find method has a time complexity of O(n/2) in the worst-**case** scenario
where n is the number of items in the list. So, the overall time complexity of the
 "remove" method is O(n/2) + O(1) = O(n/2), where O(1) represents the time taken t
o remove the item.

As a result, the average time complexity of the CDLS implementation is O(n/2).