

Assignment 3

1. Write pseudocode to swap two adjacent elements by adjusting only the links (and not the data) using [10 Points]
 - a. Singly linked list
 - b. Doubly linked list
2. Write pseudocode to implement the contains routine for MyLinkedList [5 Points]
3. The Josephus problem is the following game: N people, numbered 1 to N, are sitting in a circle. Starting at person 1, a hot potato is passed. After M passes, the person holding the potato is eliminated, the circle closes ranks, and the game continues with the person who was sitting after the eliminated person picking up the hot potato. The last remaining person wins. Thus if M=0 and N=5, players are eliminated in order and player 5 wins. If M=1 and N=5, the order of elimination is 2,4,1,5.
 - a. Write a program to solve the Josephus problem for general (integer) values of M and N. Try to make your program as efficient as possible. Make sure you dispose of cells. [7 Points]
 - b. What is the running time of your program? [3 Points]
4. What is the running time of the following code? [5 Points]

```
public static List<Integer> makeList( int N)
{
    ArrayList<Integer> lst = new ArrayList<>();
    for( inti =0; i < N; i++)
    {
        lst.add(i);
        lst.trimToSize();
    }
}
```

5. The following routine removes the first half of the list passed as a parameter: [10 Points]

```
public static void removeFirstHalf(List<?> lst)
{
    int theSize = lst.size() / 2
    for( inti =0; i < theSize; i++ )
        lst.remove(0);
}
```

- a. Why is theSize saved prior to entering the for loop?
 - b. What is the running time of removeFirstHalf if lst is an ArrayList?
 - c. What is the running time of removeFirstHalf if lst is a LinkedList?
 - d. Does using an iterator make removeFirstHalf faster for either type of List
6. Write a function in pseudocode named removeDuplicates(), which takes a singly linked list sorted in increasing order and deletes any duplicate nodes from the list. The list should only be traversed once and the routine should not call any other routine.

For example if the linked list is 11->11->11->21->43->43->60 then removeDuplicates() should convert the list to 11->21->43->60.

Input: the **head** node of the linked list .

Output:

Your function should return a pointer to the head of linked list with no duplicate element.

[10 Points]