

Assignment 2

Note: You can find JavaScript code along with your Sakai assignments. You can implement assignment problems in JavaScript or can write an Algorithm by following pseudo code format is fine. The choice is up to you.

ADT operations must use in all problems: first(), last(), isFirst(p), isLast(p), size(), isEmpty(), before(p), after(p), insertLast(e), remove(p), insertFirst(e), swapElements(p, q), insertBefore(p, e), insertAfter(p, e), replaceElement(p, e), element(), positions().
Iterators: ObjectIterator elements() with boolean hasNext(), object nextObject()

You can write additional helper methods to solve these problems.

1. Design a pseudo code algorithm to remove all duplicate elements from the Sequence. What is the time complexity of your algorithm. You can use the Sequence ADT methods in your algorithm. Example: To access the first element you can call S.first().
2. Design an algorithm, **isPermutation(A,B)** that takes two sequences A and B and determines whether or not they are permutations of each other, i.e., same elements but possibly occurring in a different order. **Hint:** assume that A and B do not contain duplicates (later we will allow duplicates).

What is the worst-case time complexity of your algorithm? Justify your answer.

Permutation: In mathematics, a permutation is an arrangement of objects in a specific order.

Example Sequence: A B C

Permutations are: A B C, A C B, B A C, B C A, C A B, C B A

Inputs: Sequence A (A B C), Sequence B(C B A) return true
Sequence A (A B) Sequence B(C B A) return false

Your algorithm needs to return true. A and B are permutations of each other.

3. DLL Practice with List ADT.
 - A. Implement the function to get the maximum value from the given Doubly Linked list.
Algorithm findMax(L)
 - B. Implement the function to return the middle of Double linked list. For a implementation hint refer Slide 23.
Algorithm findMiddle(L)

- C. Implement the function to remove the middle element in the given list.
Hint: You can use the implemented function of Task B in the Task C.
Algorithm removeMiddle(L)
- D. What is the time complexity of Task A and B.