Sukkur IBA University
Department of Computer Science

DATA STRUCTURES
Lab01 - Arrays/Linkedlist

Saif Hassan
Date: 8th January, 2020

READ IT FIRST

Prior to start solving the problems in this assignments, please give full concentration on following points.

1. WORKING - This is individual lab. If you are stuck in a problem contact your teacher, but, in mean time start doing next question (don't waste time).

2. DEADLINE - One Week after assigned this.

3. SUBMISSION - This assignment needs to be submitted in a soft copy.

4. WHERE TO SUBMIT - Please visit your Google Classroom.

5. HOW TO SUBMIT - rename all .java/.cpp files as task number and make .zip file, name it as 000-00-0000_assignment.zip, and submit .java/.cpp file ONLY.

KEEP IT WITH YOU!

1. Indent your code inside the classes and functions. It's a good practice!

2. It is not bad if you keep your code indented inside the loops, if and else blocks as well.

3. Comment your code, where it is necessary.

4. Read the entire question. Don't jump to the formula directly.

Exercises

Arrays

Task01 (NLP)

Create a file named NLArray.java and design following functions:

- **String [] wordTokenize (String fileName)** → Read any text file and return list of words from that file. (Ignore . , : and all these types operators)

- **String[] extractEmail (String fileName)** → Read any text file and return all emails from and file

Note: Read about Natural Language Processing (NLP), Word Tokenizing, Stop Words, Information Extraction/Retrieval for Knowledge.

Task02 (Image Cropping)

Design following methods in above same class NLArray.java.

- **void extractBoundaries (int arr[][])** → This function should extract boundaries and print from arr (Boundaries include 1st row, 1st col, last row, last col).

- **void cropCenterPart (int arr[][])** → This function should extract center part and print from arr, center part includes everything except Boundaries (Boundaries include 1st row, 1st col, last row, last col).

Task03 (Determining N consecutive same values)

Design following methods in above same class NLArray.java.

boolean NConRep (int arr[][]) → This function would return True if N consecutive values are same otherwise false. Check examples:
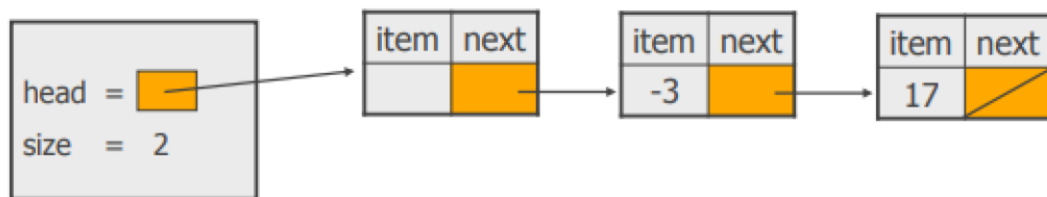
2	1	3	5
22	22	22	22
12	41	88	53
57	8	74	4

N is 4 so matrix is 4x4 and in 4 consecutive values are same so it should return True.

Task04 Linked Lists

In this task you will write a program that implements a variant of a linked list. This variant has a dummy node pointed to by the head link as shown in the following figure:

Linked list with a dummy first node:

This trick will allow your code to be a little simpler, not requiring a special case for add or remove operations. Your constructor method will be:

```
 public LinkedList(){

     head = new Node(null);

     size = 0;

}
```

You need to write a class called **LinkedList** that implements the following **List** interface:

```
// a list interface

public interface List {

 public boolean isEmpty();
      // returns true if the list is empty, false otherwise

      public int size();
      // returns the number of items in the list
       public void add(Object item);
      // adds an item to the list
      // item is added at the end of the list
       public void add(int index, Object item); //
      adds an item to the list at the given index //
      item is added at the given index; // the
      indices start from 1.

      public void remove(int index);
      // removes the item from the list that has the given index
       public void remove(Object item);
      // removes an item from the list
      // removes the first item in the list whose equal method matches
      // that of the given item

      public List duplicate();
      // creates a duplicate of the list
      // returns a copy of the linked list

      public List duplicateReversed();
      // creates a duplicate of the list with the nodes in reverse order
      // returns a copy of the linked list with the nodes in reverse order
}
```

In addition to the interface, your **LinkedList class** needs to implement a **toString()** method that prints the list in the format

[size: the_size_of_the_list - item1, item2, ...]

Specifications, notes, and hints

Your program needs to meet the following specifications:

- Submit the file `LinkedList`.java and additional files if applicable. Your `Node class` should be an inner class within the `LinkedList` class. Make sure your `class` implements the `interface` as specified, i.e. your `class` should begin with `public class LinkedList implements List`.

- When commenting your code use `Javadoc` style comments at the beginning of each method.

- Put comments at the top of the file (`Java` File) with your name, S_ID, S_Name, date and course, and a short (one or two line) description of what the program does. Make sure your code runs on machine.

Submit your `source code` files via the classroom by the due date (remember the course syllabus for the late policy).