Krissanapong Palakham sec.1 6288102

ITCS231 Data Structures & Algorithm Analysis
Lab Exercise 3

Faculty of ICT, Mahidol University
September 18, 2020

Objectives

- This lab exercise covers **FOUR** lectures: Linked List, Stack, Advance Sorting and Recursive.
- The purpose of the exercise is to have you familiar with different data structures and algorithms.

Introduction

Info: There are **Six** tasks in this lab exercise to complete. You have to create **Five** Java files entitled with lab03TaskX.java, where 'X' is task number and follow the instruction. Task no.6, the last one, is try running the given program and answers to the questions.

1 Double-Ended Linked List

You are given **FirstLastList.class (**firstLastList.java) written by Robert Lafore. Create a Java file named lab03Task1.java and write a code to do the following:

- Creates a new object of FirstLastList class.
- Inserts 5, 15, 25, and 30 at the beginning of linked list.
- Inserts 3, 13, 23, and 33 at the end of linked list.
- Deletes and show the deleted link on the screen (until the linked lists is empty).

2 Doubly Linked List

You are given **DoublyLinkedList.class (**doublyLinked.java) written by Robert Lafore. Create a Java file named lab03Task2.java and write a code to do the following:

- Creates a new object of DoublyLinkedList class.
- Inserts 5, 15, 25, and 30 at the beginning of linked list.
- Inserts 3, 13, 23, and 33 at the end of linked list.
- Inserts 100 after 23.
- Deletes the First element
- Deletes the Last element.
- Delete 25.

• Display the list forwardly.
• Display the list backwardly.

3 Stack by Linked List

You are given **LinkStack.class (**linkStack..java) written by Robert Lafore. Create a Java file named lab03Task3.java
and write a code to do the following:

• Creates a new object of LinkStack.
• Pushes 16, 25, 36, and 49.
• Pops and show the result on the screen (until the list is empty). on the screen.

4 Merge Sort

You are given **DArray.class** (mergeSort.java) written by Robert Lafore. Create a Java file named lab03Task4.java and write a code to do the following:

• Creates a new object of DArray class using 50 as a parameter for constructor.
• Inserts 11, 12, 8, 5, 3, 15, 20, 18, 36, and 17.
• Sorts the array and show the result on screen.

5 Binary Search

You are given a **binarySearch.java** written by Robert Lafore. This program performs binary search for a given search key. Create a Java file named **lab03Task5.java** by modifying the given program to count for number of search key comparisons used in searching. Compile and run your program with given input search key, observe the results and answer to the questions as follows.

(a) Input search key : **135**
Searching is found or not found ? How many comparisons used ? 7

(b) Input search key : **98**
Searching is found or not found ? How many comparisons used ? 16

(c) Input search key : **64**
Searching is found or not found ? How many comparisons used ? 25

6 Anagram

You are given an **AnagramApp.java** written by Robert Lafore. This program performs anagramming on word. Compile and run this program with given input words, observe the results and answer the questions.

(a) Input word : **birds**

How many possible words generated by the program ? 120

How many permutations calculated by using formula* ? n! = 5! = 120

Are the 2 answers equal ? Why/Why not ? Yes, because we're using recursive function. First, we rotate word with last 2 characters, then add to 3 characters, 4 characters, and 5 characters, so we will get equal answer.

(b) Input word : **balls**

How many possible words generated by the program ? 120

How many permutations calculated by using formula* ? 60

Are the 2 answers equal ? Why/Why not ? No, because in our code we didn't write a code when the word has repeated character.

* The formula to calculate permutation is (ref. ITCS320 Discrete Structures, topic "Counting")

$$\mathbf{p = n! / (k_1! \, k_2! k_3! \, \ldots k_m!),}$$

where **p** is total number of permutations,
n is word length (in characters),
$\mathbf{k_i}$ is number of times each character repeats in a word.