EC4070: Data Structures and Algorithms

LAB 02

K.M.J.G.S.C.W BANDARA

2021/E/073

SEMESTER 4

EC4070

04.10.2023

```
C:\Users\SASINDU\Desktop\lab2>javac Q2.java

C:\Users\SASINDU\Desktop\lab2>java Q2
Enter the how many elements in your array: 6
Enter the elements in sorted order:
32
76
5
38
42
13
Enter the target value: 5
Element 5 found at index 2
```

C:\Users\SASINDU\Desktop\lab2\b_S_Recursive.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

RecursiveProblems.java | Q2.java | b_S_Recursive.java

```java


        System.out.println("Enter the elements of the sorted array:");

        for (int i = 0; i < size; i++) {

            arr[i] = scanner.nextInt();

        }



        System.out.print("Enter the target element: ");

        int trg = scanner.nextInt();



        int result = b_S_Recursive(arr, trg, 0, size - 1);



        if (result != -1) {

            System.out.println("Element found at index: " + result);

        } else {

            System.out.println("Element not found in the array.");

        }



        scanner.close();

    }

}
```

```
C:\Users\SASINDU\Desktop\lab2>javac b_S_Recursive.java

C:\Users\SASINDU\Desktop\lab2>java b_S_Recursive
Enter array size: 7
Enter the elements of the sorted array:
32
54
14
8
19
5
31
Enter the target element: 4
Element not found in the array.
```

```
C:\Windows\system32\cmd.e:

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SASINDU\Desktop\lab2>javac RecursiveProblems.java

C:\Users\SASINDU\Desktop\lab2>java RecursiveProblems
Enter triangular number: 3
Triangular number of 3 is 6
Enter a number to get factorial: 4
Factorial of 4 is 24
Enter the 1st string: car
Enter the 2nd string: wash
car and wash are not anagrams.
Enter the number of disks for Towers of Hanoi: 6
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
```

C:\Users\SASINDU\Desktop\lab2\RecursiveProblems.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

RecursiveProblems.java

```java
    public static int cal_Factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * cal_Factorial(n - 1);
        }
    }

    //  Anagrams
    public static boolean areAnagrams(String str1, String str2) {
        str1 = str1.toLowerCase();
        str2 = str2.toLowerCase();


        if (str1.length() != str2.length()) {
            return false;
        }

        if (str1.length() == 0) {
            return true;
        }

        char firstChar = str1.charAt(0);
        int index = str2.indexOf(firstChar);

        if (index == -1) {
            return false;
        } else {
            String newStr1 = str1.substring(1);
            String newStr2 = str2.substring(0, index) + str2.substring(index + 1);
            return areAnagrams(newStr1, newStr2);
        }
    }

    //  Towers of Hanoi
    public static void towers_Hanoi(int numDisks, char source, char destination, char auxiliary) {
        if (numDisks == 1) {
            System.out.println("Move disk 1 from " + source + " to " + destination);
            return;
        }

        towers_Hanoi(numDisks - 1, source, auxiliary, destination);
        System.out.println("Move disk " + numDisks + " from " + source + " to " + destination);
        towers_Hanoi(numDisks - 1, auxiliary, destination, source);
    }
}
```

C:\Users\SASINDU\Desktop\lab2\b_S_Recursive.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

RecursiveProblems.java | Q2.java | b_S_Recursive.java

```java
import java.util.Scanner;

public class b_S_Recursive {

    public static int b_S_Recursive(int[] arr, int trg, int LHS, int RHS) {

        if (LHS <= RHS) {

            int mid = LHS + (RHS - LHS) / 2;



            if (arr[mid] == trg) {

                return mid;

            } else if (arr[mid] < trg) {

                return b_S_Recursive(arr, trg, mid + 1, RHS);

            } else {

                return b_S_Recursive(arr, trg, LHS, mid - 1);

            }

        }


        return -1;

    }


    public static void main(String[] args) {

         Scanner scanner = new Scanner(System.in);


        System.out.print("Enter array size: ");

        int size = scanner.nextInt();


        int[] arr = new int[size];


        System.out.println("Enter the elements of the sorted array:");
```

Java source file | length : 1,408 lines : 90 | Ln : 34 Col : 6 Pos : 554

C:\Users\SASINDU\Desktop\lab2\Q2.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

RecursiveProblems.java | Q2.java

```java
import java.util.Scanner;

public class Q2 {

    public static int function1(int[] arr, int t) {

        int LHS = 0;

        int RHS = arr.length - 1;


        while (LHS <= RHS) {

            int mid = LHS + (RHS - LHS) / 2;

            if (arr[mid] == t) {

                return mid;

            } else if (arr[mid] < t) {

                LHS = mid + 1;

            } else {

                RHS = mid - 1;

            }

        }

        return -1;

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the how many elements in your array: ");

        int n = scanner.nextInt();

        int[] arr = new int[n];

        System.out.println("Enter the elements in sorted order:");


        for (int i = 0; i < n; i++) {

            arr[i] = scanner.nextInt();

        }

        System.out.print("Enter the target value: ");

        int t = scanner.nextInt();

        int sum = function1(arr, t);


        if (sum != -1) {

            System.out.println("Element " + t + " found at index " + sum);

        } else {

            System.out.println("Element " + t + " not found in the array.");

        }

        scanner.close();

    }
```

Java source file | length : 1,327 lines : 81 | Ln : 19 Col : 28 Pos : 311

C:\Users\SASINDU\Desktop\lab2\RecursiveProblems.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

RecursiveProblems.java

```java
import java.util.Scanner;

public class RecursiveProblems {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        //  Triangular Number
        System.out.print("Enter triangular number: ");
        int tNI = scanner.nextInt();
        int tNR = cal_Triangular_Number(tNI);
        System.out.println("Triangular number of " + tNI + " is " + tNR);

        //  Factorial
        System.out.print("Enter a number to get factorial: ");
        int fI = scanner.nextInt();
        int fR = cal_Factorial(fI);
        System.out.println("Factorial of " + fI + " is " + fR);

        scanner.nextLine();

         // Anagrams
        System.out.print("Enter the 1st string: ");
        String str1 = scanner.nextLine();
        System.out.print("Enter the 2nd string: ");
        String str2 = scanner.nextLine();

        boolean areAnagrams = areAnagrams(str1, str2);
        if (areAnagrams) {
            System.out.println(str1 + " and " + str2 + " are anagrams.");
        } else {
            System.out.println(str1 + " and " + str2 + " are not anagrams.");
        }

        //  Towers of Hanoi
        System.out.print("Enter the number of disks for Towers of Hanoi: ");
        int numDisks = scanner.nextInt();
        towers_Hanoi(numDisks, 'A', 'C', 'B');

        scanner.close();
    }

    //  Triangular Number
    public static int cal_Triangular_Number(int n) {
        if (n <= 0) {
            return 0;
        } else {
            return n + cal_Triangular_Number(n - 1);
        }
    }

    //  Factorial
    public static int cal_Factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
```

Java source file | length : 3,042 lines : 102 | Ln