GTU Department of Computer Engineering

CSE 222 / 505 – Spring 2022

Homework 4

Ahmet USLUOGLU

# 1801042602

### – System Requirement

Operating System must have JDK (Java Development Kit) 11 and JRE (Java Runtime Environment) 11 or higher.

There should be enough space for storing data’s.

1. – Class Diagrams

No Class diagram.

3. Problem - Solution Approach

Problem:

**Q1**. Write a recursive function to search a given string in another given bigger string. The function should return the index of the i’th occurrence of the query string and return -1 when the query string doesn’t occur in the big string, or the number of occurrences is less than i.

Solution:

To be able to find the n’th occurrence of a substring in a string I have used String class indexOf method. indexOf method returns the index of the first occurrence of a sub string in the main string. After finding the first occurrence I called my function recursively and set the starting index the first occurrence’s index + 1. At each iteration if indexOf method finds a value greater than -1 I increased the count. When count becomes equal to occurrence value function returns the last index it found.

Problem:

**Q2**. Suppose that you are given a sorted integer array. Suggest a recursive algorithm to find the number of items in the array between two given integer values.

Solution:

To find the number of items in between given integers I have used a similar algorithm to the binary search algorithm. If the middle element of the array is bigger than the upper border function recalls itself recursively to look for the left of the array which has smaller numbers. If the middle element of the array is smaller than the lower border function recalls itself recursively to look for the right of the array which has bigger numbers. If the number is in between borders function recursively calls itself to look for the right and left of the array while adding 1 to return statement. Every time an element is in bounds it recalls the function. When the arrays left and right indexes are equal

Function returns 1. After all recursive calls added to each other total is equal to number of items in between the borders. If there is no element in between borders function returns 0;

Problem:

**Q3**. Suppose that you are given an unsorted integer array. Propose a recursive solution to find contiguous subarray/s that the sum of its/theirs items is equal to a given integer value.

Solution:

When the program starts function checks if the 0th index element is bigger than the target number. If it is smaller than the target number functions subtracts the number at the index from target number and keeps it as remain.

If the number at the index is bigger than the remained number function recursively calls itself and stars from the index + 1 that it’s started at the beginning the functions reaches the end of the array it increases the next starting index by one and recalls itself to start again.

Problem:

**Q5**. Suppose that you are given a 1-D array of empty blocks of length L. Propose a recursive algorithm that calculates all the possible configurations to fill this array with colored-blocks with length at least 3 (i. e., the length of colored blocks can be from 3 to L). Note that there should be at least one empty block between each consecutive colored-block.

Solution:

Starts with 3 block size and moves one index at a time. To calculate all possibilities After every calculation recalls itself and sends the filled array from the index +1 that is filled.