

# 1. System Requirements

Firstly, in my code, most important thing is Question classes. I create Question objects for each question to perform necessary calculations.

In Q1, we need two string to find small string in big string.

In Q2, there should be integer array and it must be sorted.

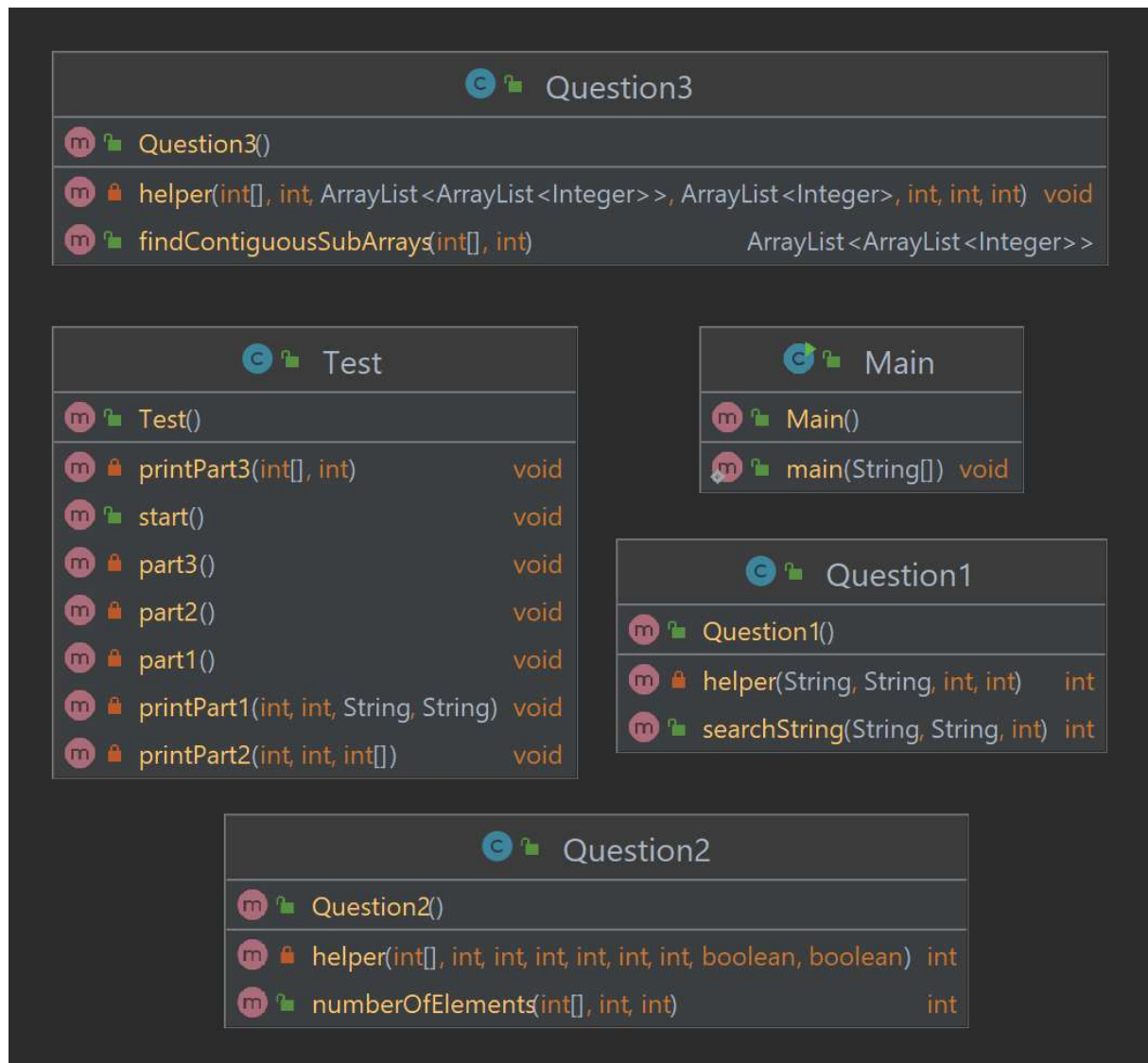
In Q3, we just need integer array and target to find subarrays that their sum is target.

We also need Operating system that have jdk 8 and jre 8 for use the program.

We need some space to store objects of Question1, Question2, Question3 and strings/arrays. There is many integer array and string to perform test case.

Other thing we should consider is our resources like CPU, RAM, Stack size etc. Recursion method uses much more resources than loop method. So better CPU gives faster result.

## 2. Class Diagram



### 3. Problem Solutions Approach

Our main aim in this homework is using recursive method correctly and effectively. So, before I start to do questions, I was think about it and I write templates of questions and their pseudocode to paper. Then I started to coding.

Firstly, I write base cases of recursive methods. I asked myself that “in which case this recursive method should end processing”. After that I write sub-cases and lastly recursive method body. With this approach, questions didn't take a long time.

I faced many of exceptions like `ArrayOutOfBound`, `StackOverflow` etc. But I know what does exception mean. I correct the errors one after another.

After that, I create Test class to perform some tests on methods. Also exception handling is performed. In test program, every possibilities are tested.

After all, I worked on time complexities of questions. Time complexities are calculated and proved 2 of time complexities with induction method.

## 4. Test Cases

### Question-1:

```
int index;
int occurrence1 = 1;
int occurrence2 = 2;
int occurrence3 = 3;
int occurrence4 = 4;
int occurrence5 = 5;

String bigString1 = "ayusufbcyusufdefyusufghijklkyusuf";
String queryString1 = "yusuf";
String bigString2 = " ";
String queryString2 = "deneme";
String bigString3 = "question asdf ghj question76267 question";
String queryString3 = "question";

index = q1.searchString(queryString1, bigString1, occurrence1);
printPart1(index, occurrence1, bigString1, queryString1);

index = q1.searchString(queryString1, bigString1, occurrence2);
printPart1(index, occurrence2, bigString1, queryString1);

index = q1.searchString(queryString1, bigString1, occurrence5);
printPart1(index, occurrence5, bigString1, queryString1);

index = q1.searchString(queryString2, bigString2, occurrence1);
printPart1(index, occurrence1, bigString1, queryString1);

index = q1.searchString(queryString3, bigString3, occurrence1);
printPart1(index, occurrence1, bigString3, queryString3);

index = q1.searchString(queryString3, bigString3, occurrence4);
printPart1(index, occurrence4, bigString3, queryString3);
```

-----\_QUESTION-1\_-----

1th occurrence of yusuf founded in ayusufbcyusufdefyusufghijklyusuf at index 1

2th occurrence of yusuf founded in ayusufbcyusufdefyusufghijklyusuf at index 8

There is no 5 times of yusuf in the ayusufbcyusufdefyusufghijklyusuf

There is no 1 times of yusuf in the ayusufbcyusufdefyusufghijklyusuf

1th occurrence of question founded in question asdf ghj question76267 question at index 0

There is no 4 times of question in the question asdf ghj question76267 question

## Question-2:

```
private void part2(){
    int[] sortedArray1 = new int[]{10,20,30,40,50,60,70,80,90};
    int[] sortedArray2 = new int[]{2, 4, 6, 8, 10, 12, 14, 16};
    int[] sortedArray3 = new int[]{53, 54, 55, 61, 81, 153, 155, 161, 167};
    int first1 = 20, second1 = 70, first2 = 4, second2 = 4, first3 = 54, second3 = 55, first4 = 81, second4 = 167;
    printPart2(first1, second1, sortedArray1);
    printPart2(first2, second2, sortedArray2);
    printPart2(first3, second3, sortedArray3);
    printPart2(first4, second4, sortedArray3);
}

private void printPart2(int first, int second, int[] arr){
    Question2 q2 = new Question2();
    if(first == second){
        System.out.println("There is 0 element between "+first+" and "+second+"\n");
    }
    else{
        int number = q2.numberOfElements(arr,first,second);
        if(number == -1)
            System.out.println("One or more element doesn't exist in array!\n");
        else{
            System.out.println("There is "+number+" elements between "+first+" and "+second+"\n");
        }
    }
}
```

-----\_QUESTION-2\_-----

There is 4 elements between 20 and 70

There is 0 element between 4 and 4

There is 0 elements between 54 and 55

There is 3 elements between 81 and 167

### Question-3:

```
private void part3(){
    int[] elements = new int[]{9, 1, 3, 6, 4, -2, 8, 7, 9, -6, 10};
    int target = 10;
    int[] elements1 = new int[]{1, 4, 5, -8, 2, 7, 9, 1, -5, 6, 3, };
    int target1 = 5;
    int[] elements2 = new int[]{2, 4, 6 -5, 8, -3, 10, 1, -2, 3, 5, 7, -6};
    int target2 = 6;
    printPart3(elements, target);
    printPart3(elements1, target1);
    printPart3(elements2, target2);
}

private void printPart3(int[] elements, int target){
    Question3 q3 = new Question3();
    ArrayList<ArrayList<Integer>> arrays;
    arrays = q3.findContiguousSubArrays(elements, target);
    System.out.println("My elements are : ");
    for (int i = 0 ; i < elements.length ; i++){
        System.out.print "["+elements[i]+" " );
    }
    System.out.println("\nFounded Integers that their sum is "+target+" :");
    for (ArrayList<Integer> array : arrays) {
        for (Integer integer : array) {
            System.out.print(integer + " ");
        }
        System.out.println();
    }
    System.out.println();
}
}
```

-----QUESTION-3-----

My elements are :

[9] [1] [3] [6] [4] [-2] [8] [7] [9] [-6] [10]

Founded Integers that their sum is 10 :

9 1

1 3 6

6 4

4 -2 8

7 9 -6

10

My elements are :

[1] [4] [5] [-8] [2] [7] [9] [1] [-5] [6] [3]

Founded Integers that their sum is 5 :

1 4

5

9 1 -5

1 -5 6 3

My elements are :

[2] [4] [1] [8] [-3] [10] [1] [-2] [3] [5] [7] [-6]

Founded Integers that their sum is 6 :

2 4

1 8 -3

-3 10 1 -2

-2 3 5

5 7 -6

## 5. Time Complexities

### Question-1:

$$T(n) = 1 \quad n=0;$$

$$T(n) = T(n-1) + 1 \quad n>1;$$

Using substitution method,  $T(n) = \theta(n)$ .

Prove with Induction method:

For  $n = 0$ , its true.

Assume  $n = k$  is true.  $T(k) = \theta(k)$

For  $n = k+1$ ,

It must be  $T(k+1) = \theta(k+1)$

$$T(k+1) = T(k) + 1 \Rightarrow \theta(k) + 1$$

So, it means it's TRUE.

### Question-2:

$T(n) = T(n/2) + T(n/2) + 1 \Rightarrow$  There is 2 binary search in the method.

$$T(n) = 2T(n/2) + 1$$

$$= 2T(n/2) + k$$

$$= 2\{2T(n/4) + k\} + k$$

$$= 4T(n/4) + 3k$$

$$= \dots$$

$$= n.T(1) + (n-1)k$$

$$= n.k + (n-1)k$$

$$= 2nk - k$$

$$= O(n).$$

So, using substitution method  $T(n) = O(n)$ .



### Question-3:

$$T(n) = 1 \quad n = 0$$

$$T(n) = T(n-1) + n \quad n > 0$$

$$T(n-1) = T(n-2) + n + (n-1)$$

$$T(n-2) = T(n-3) + n + (n-1) + (n-2)$$

$$T(n-3) = T(n-4) + n + (n-1) + (n-2) + (n-3)$$

....

$$T(2) = T(1) + n + (n-1) + ..... + 2$$

$$T(1) = T(0) + n + (n-1) + ..... + 2 + 1$$

$$\Rightarrow n + (n-1) + (n-2) + (n-3) + ..... + 2 + 1 = \frac{n*(n+1)}{2}$$

$$\text{So, } T(n) = \frac{n*(n+1)}{2} \Rightarrow T(n) = O(n^2)$$

Prove with Induction method:

For  $n = 0$ , its true.

Assume  $n = k$  is true.  $T(k) = O(k^2)$

For  $n = k+1$ ,

It must be  $T(k+1) = O((k+1)^2)$

$$T(k+1) = T(k) + k + 1 \Rightarrow O(k^2) + k + 1$$

So, it means it's TRUE.

## 6. Running Command and Results

The file should be unzipped and opened as a Project in IntelliJIdea. All the methods are accessible on driver code.