GTU Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework #04 Report

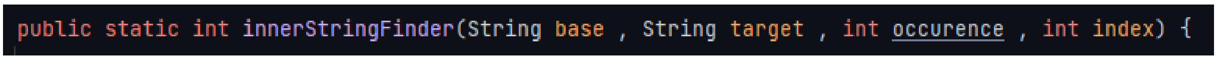
Ali Kaya

1901042618

a.kaya2019@gtu.edu.tr

1.Detailed System Requirements

#For Q1:

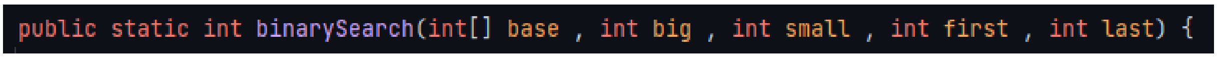


```
public static int innerStringFinder(String base , String target , int occurence , int index) {
```

User need to give three parameter:

1) String base is the big String we are searching in it.
2) String target is the target string we will search inside the big string.
3) İnt occurence is the number specifying the number of occurrences whose index we are looking for.
4) İnt index used for recursive calls inside the function,initial value must be 0

#For Q2:

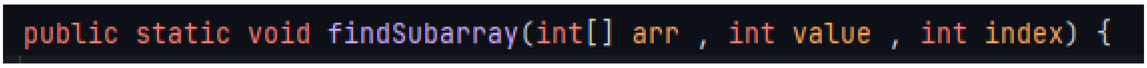


```
public static int binarySearch(int[] base , int big , int small , int first , int last) {
```

User need to give five parameter:

1) İnt[] base is the array we searched in it.
2) İnt big specifies the upper limit of the numbers searched.
3) İnt small specifies the lower limit of the numbers searched.
4) İnt first used for recursive calls inside the binary search algorithm,initial value must be 0.
5) İnt last used for recursive calls inside the binary search algorithm,initial value must be base array's length-1.

#For Q3:

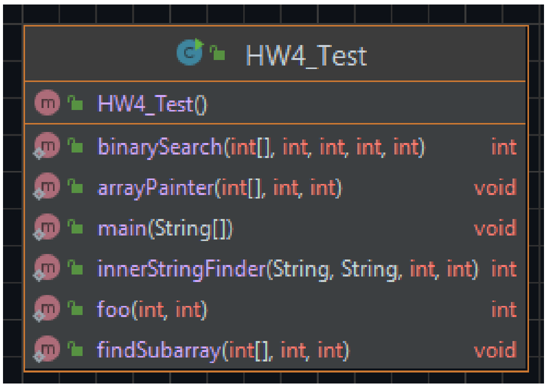


```
public static void findSubarray(int[] arr , int value , int index) {
```

User need to give three parameter:

1) İnt[]arr is the array we searched in it.
2) İnt value the number to be compared with the numbers summed
3) İnt index used for recursive calls inside the function,initial value must be 0

2. Class Diagrams



HW4_Test

HW4_Test()	
binarySearch(int[], int, int, int, int)	int
arrayPainter(int[], int, int)	void
main(String[])	void
innerStringFinder(String, String, int, int)	int
foo(int, int)	int
findSubarray(int[], int, int)	void

3. Problem solutions approach

#For Q1:

Each time with recursive call, I increase index by one and compare the target string with the string in the range from index to (index + target.length()). If it is the same, I decrease the occurrence by one if not i increase the index number by one again and call recursive call, then if the occurrence count is 0, I return the index.

Complexity Analysis => recurrence relation: T(N) = 2T(N-1) + theta(N)

T(N) = theta(N*(2^N))

#For Q2:

Each time with recursive call I compare the value in the middle of the sub array with the target value each time, if it's the same, I add a value to the return before the recursive call, otherwise I decide whether the middle value is large or small and decide which subarray to look at, then I make a recursive call again.

Complexity Analysis => recurrence relation: T(n) = 2T(n/2) + 1 , T(1) = log n

T(N) = O(n log n)

#For Q3:

I am using the for loop to sum specific indices inside the array so this adds theta(N) to time complexity . If the result of the recursive call in the for is equal to my target value, I print the first and last indices of the subarray to the screen. Then I make a recursive call again.

Complexity Analysis => recurrence relation: T(N) = T(N-1) + theta(N)

T(N) = theta(N^2)

#For Q4:

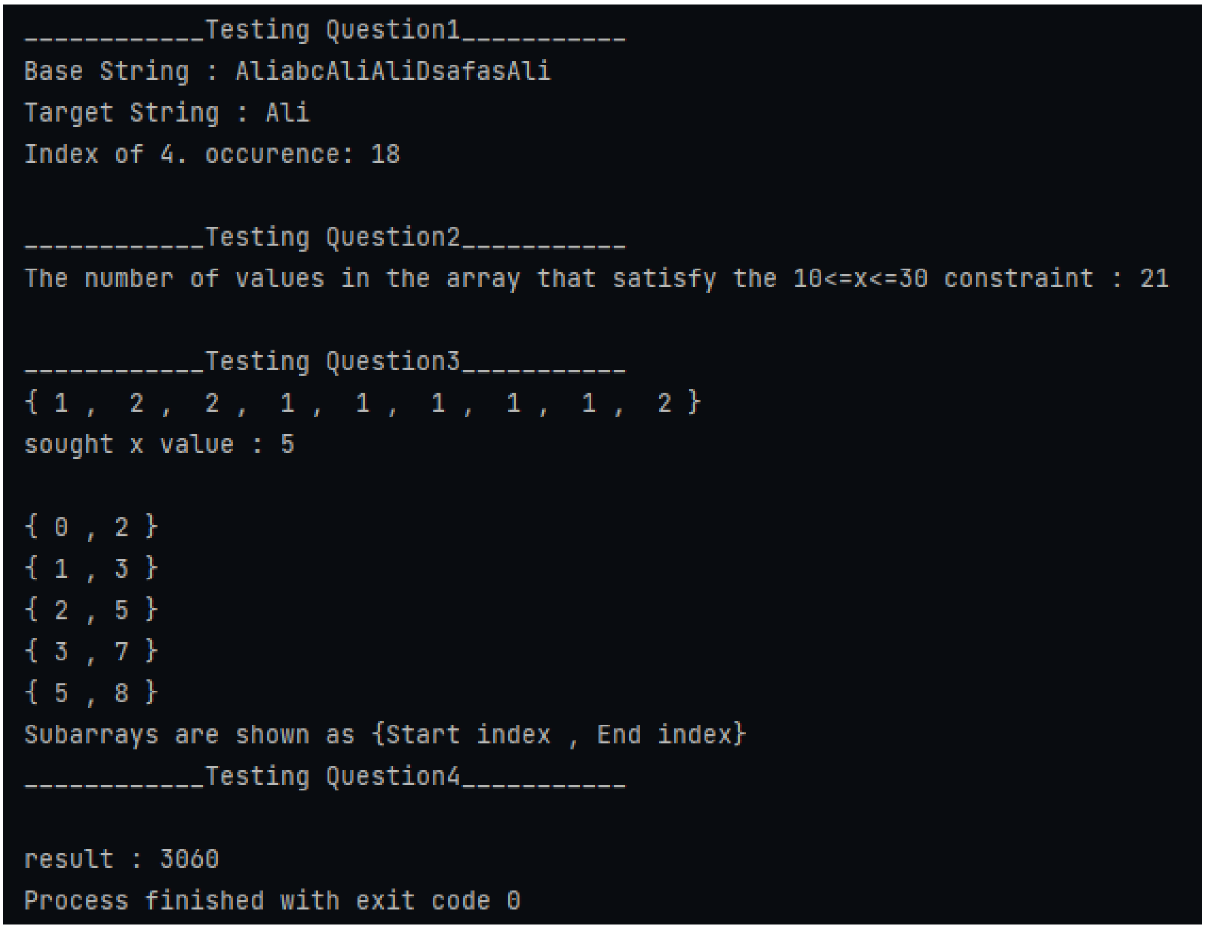
There are number of recursive calls depending on the smaller integer

The number of multiplication operations which is the basic operation determines our Time Complexity

Complexity Analysis =>   T(n) = log^2n * 2^n

Function dividing integers in half and calling recursive again “logns” comes from there.

4)Test Cases and Running command and results



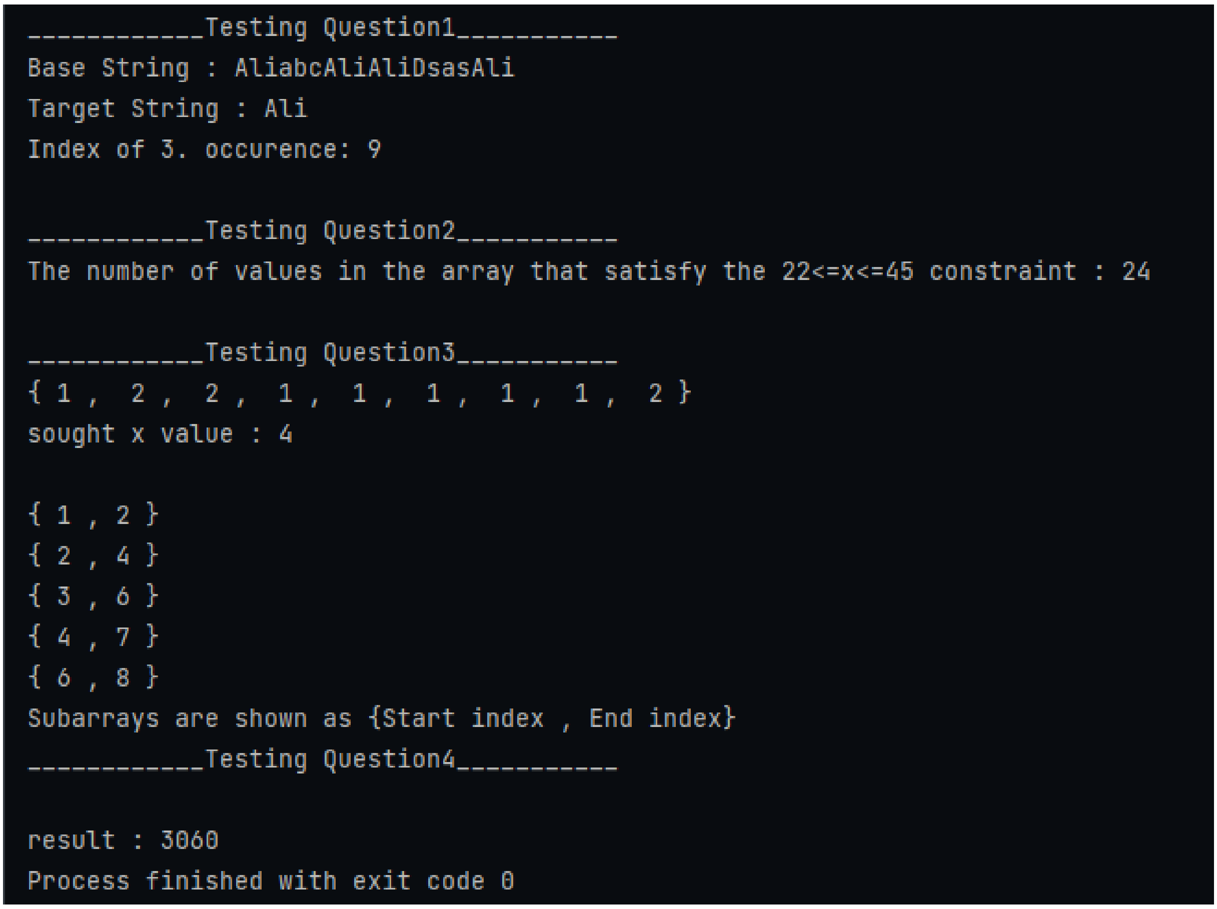
```
____________Testing Question1___________
Base String : AliabcAliAliDsafasAli
Target String : Ali
Index of 4. occurence: 18

____________Testing Question2___________
The number of values in the array that satisfy the 10<=x<=30 constraint : 21

____________Testing Question3___________
{ 1 ,  2 ,  2 ,  1 ,  1 ,  1 ,  1 ,  1 ,  2 }
sought x value : 5

{ 0 , 2 }
{ 1 , 3 }
{ 2 , 5 }
{ 3 , 7 }
{ 5 , 8 }
Subarrays are shown as {Start index , End index}
____________Testing Question4___________

result : 3060
Process finished with exit code 0
```



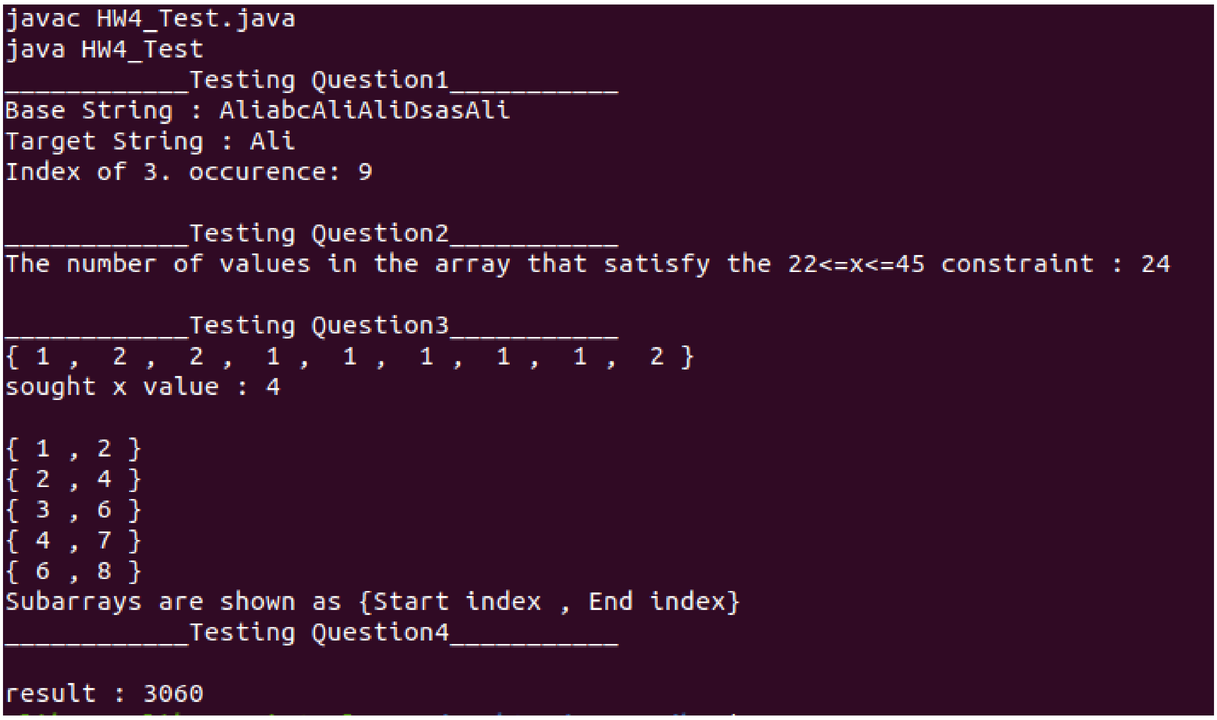
```
____________Testing Question1___________
Base String : AliabcAliAliDsasAli
Target String : Ali
Index of 3. occurence: 9

____________Testing Question2___________
The number of values in the array that satisfy the 22<=x<=45 constraint : 24

____________Testing Question3___________
{ 1 ,  2 ,  2 ,  1 ,  1 ,  1 ,  1 ,  1 ,  2 }
sought x value : 4

{ 1 , 2 }
{ 2 , 4 }
{ 3 , 6 }
{ 4 , 7 }
{ 6 , 8 }
Subarrays are shown as {Start index , End index}
____________Testing Question4___________

result : 3060
Process finished with exit code 0
```



```
javac HW4_Test.java
java HW4_Test
____________Testing Question1___________
Base String : AliabcAliAliDsasAli
Target String : Ali
Index of 3. occurence: 9

____________Testing Question2___________
The number of values in the array that satisfy the 22<=x<=45 constraint : 24

____________Testing Question3___________
{ 1 ,  2 ,  2 ,  1 ,  1 ,  1 ,  1 ,  1 ,  2 }
sought x value : 4

{ 1 , 2 }
{ 2 , 4 }
{ 3 , 6 }
{ 4 , 7 }
{ 6 , 8 }
Subarrays are shown as {Start index , End index}
____________Testing Question4___________

result : 3060
```