Algorithm Analysis in Java

Objective:
The objective of this lab exercise is to implement recursive algorithms for various problems. You will implement three different recursive algorithms and measure their performance on various input sizes.

Algorithms to Implement:
1. All binary strings without consecutive 1's: Given a constant number k, print all the binary numbers up to 2^(k) – 1, in string length of k that do not have consecutive "1"s.

    Input k: 3
    Output: 000, 001, 010, 100, 101,
    Input k:4
    Output: 0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010,

2. Show all the combinational strings of length r from an array: Given an input of n length of an integer array and a constant r value. Print all possible strings of elements in size of "r" from the given array.

    Input: arr=[1, 3, 5, 4] r = 3
    Output: 135 154 354
    Input: arr=[2, 4, 6, 1] r= 2
    Output: 26 21 46 41 61

```
||1354||
||1||   ||3||   ||5||
||13|| ||15||          ||35||
||135|| ||134||       ||154||          ||354||
```

3. First character x in a string: Take an input of a character x from the user and given the string, recursively find the occurrence of its position:

    Input: str = DataStructuresRecursion, x= 't'
    Output: 2
    Input: str = DataStructuresRecursion, x = 'R'
    Output: 14

Instructions:
1. Create a Java class that contains the three sorting algorithms (BinaryStringCons, ArrayCombination, and CharOccur).
2. Implement each recursive algorithm in separate methods within the class.
3. Implement a method that calculates and prints the time taken to sort an array using each of the sorting algorithms.
4. Run the algorithms on arrays of different sizes (e.g., 100, 1000, 10000, 100000 elements) by generating input randomly.
6. Record and analyze the time taken for sorting on each input size.

Hints:
- You can use the `System.currentTimeMillis()` or `System.nanoTime()` methods to measure the execution time of your sorting algorithms.

Sample Code:

```
import java.util.Random;

public class RecursiveLab {
  public static void main(String args[]) {
    RecursiveLab lab = new RecursiveLab();
    lab.runRecursions();
  }
  public static String toString(char[] a) {
    String string = new String(a);
    return string;
  }
  public static void BinaryStringCons (int k) {

   //1st Algorithm
  }
  public static void generate(int k, char[] ch, int n) {
    //1st Algorithm helper

  }

  static void ArrayCombination(int arr[], int r)
  {
    //2nd Algorithm
  }


  static void combinationUtil(int arr[], int data[], int start,
                  int end, int index, int r)
  {
    //2nd Algorithm helper
  }
  public static void CharOccur (String str, char x) {
           //3rd Algorithm
  }

  public static int Occur(String str, char x, int idx){
    //4th Algorithm
  }
```

```
    public static void runRecursions () {


        // Perform recursions
        // Measure and compare the time taken for the recursion algorithms for different input
sizes
    }
}
```