

三：字符串循环和集合的使用

- 对集合的考察集中在集合的特性和功能。set唯一性。list有序性。以及集合元素的个数。

2019java C组第三题

I

- 对集合的考察集中在集合的特性和功能。set唯一性。list有序性。以及集合元素的个数。

2019java C组第三题

一个字符串的非空子串是指字符串中长度至少为1的连续的一段字符组成的串。例如，字符串 aaab 有非空子串 a, b, aa, ab, aaa, aab, aaab 一共 7 个。注意在计算时，只算本质不同的串的个数。

请问，字符串 0100110001010001 有多少个不同的非空子串？

一个字符串的非空子串是指字符串中长度至少为1的连续的一段字符组成的串。例如，字符串 aaa 有非空子串 a, b, aa, ab, aaa, aab, aaab 一共 7 个。注意在计算时，只算本质不同的串的个数。

请问，字符串 0100110001010001 有多少个不同的非空子串？

分析：set的唯一性，去除重复的子串

- aaab从第一个下标开始截取: a aa aaa aaab
- 从第二个下标开始: a aa aab
-
- 下标0 截取的范围: [0,最大下标].
- 下标1的截取范围: [1,最大下标]
-
- 截取的方法: subString(i,j):截取子串

提示:

add(Object o) : 将指定元素添加到 **Set** 中 (不会添加重复的)

remove(Object o): 将指定元素从 **Set** 中删除

contains(Object o): 判断 **Set** 中是否包含指定元素

size(): 返回 **Set** 中的元素数量

发送至: 所有人
请输入消息...

记得
程
知
李
能
郑
set
姚
set
发
清

```

import java.util.HashSet;
import java.util.Set;
public class Main {
    public static void main(String[] args) {
        Set<String> set = new HashSet<>();
        String s = "0100110001010001";
        for(int i = 0; i < s.length(); i++) {
            for(int j = i; j < s.length(); j++) { //j必须从i开始
                String s1 = s.substring(i, j+1);
                set.add(s1); //不会加入重复的
            }
        }
        System.out.println(set.size()); //返回set的大小即为答案
    }
}

```

问题描述

我们知道第一个质数是 2、第二个质数是 3、第三个质数是 5.....请你计算 第 2019 个质数是多少？

答案提交

这是一道结果填空的题，你只需要算出结果后提交即可。本题的结果为一个整数，在提交答案时只填写这个整数，填写多余的内容将无法得分。

17569

```

package java1;
import java.util.HashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        int count = 0;
        int number = 2;
        while (true) { //循环直到找到2019个素数
            if (IsPrime(number)) {
                count++;
            }
            if (count == 2019) {
                break;
            }
            number++;
        }
    }
}

```

```
    }  
    System.out.println(number);  
  
}  
public static boolean IsPrime(int n) { //判断素数  
    if(n <= 3) {  
        return n > 1;  
    }  
    for(int i = 2; i < n; i++) {  
        if(n % i == 0) return false;  
    }  
    return true;  
}  
}
```