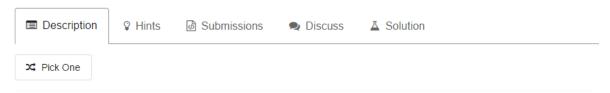
187. Repeated DNA Sequences



All DNA is composed of a series of nucleotides abbreviated as A, C, G, and T, for example: "ACGAATTCCG". When studying DNA, it is sometimes useful to identify repeated sequences within the DNA.

Write a function to find all the 10-letter-long sequences (substrings) that occur more than once in a DNA molecule.

Example:

}

```
Input: s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"
Output: ["AAAAACCCCC", "CCCCCAAAAA"]
```

题意:给出一个字符串,其中只有A,C,G,T四个字母,每10个字母作为一个子字符串,要求找到出现不止一次的子字符串。这道题直接用hashTable方法求解,遍历字符串,对每个子字符串做判断,若在hashTable中不存在,就添加进去;若存在,如果出现的次数为1,那么添加进结果中,并更新出现次数,否则继续遍历。

```
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.List;
public class L187 {
    public List<String> findRepeatedDnaSequences(String s) {
        List<String> res = new ArrayList<String>();
        Hashtable<String, Integer> temp = new Hashtable<String, Integer>();
    //将每一个长度为10的子字符串进行遍历,没有就其放进hashtable里面,出现一次就添加进结果中。
        for(int i = 0; i < s.length() - 9; i++) {</pre>
           String subString = s.substring(i,i+10);
            //这儿不是contains, 而是<u>containskey</u>, contains(value)意思是测试此映射表是否存在与指定值相关联的键,如果有则返回true。
            if(temp.containsKey(subString)){
                int count = temp.get(subString);
               if(count == 1){//如果为1,则添加结果,并且将次数设为2,当它大于1时,不需要进行处理,防止重复加入
                   temp.remove(subString);
                   temp.put(subString, 2);
                   res.add(subString);
               }
            }else {
               temp.put(subString, 1);
        return res;
```