1. 题目
2. 题干

给定一个非空的字符串，判断它是否可以由它的一个子串重复多次构成。给定的字符串只含有小写英文字母，并且长度不超过10000

1. 示例

示例 1:

输入: "abab"

输出: True

解释: 可由子字符串 "ab" 重复两次构成。

示例 2:

输入: "aba"

输出: False

示例 3:

输入: "abcabcabcabc"

输出: True

解释: 可由子字符串 "abc" 重复四次构成。 (或者子字符串 "abcabc" 重复两次构成。)

1. 题解
2. 思路

判断非空字符串是否由多个重复子串构成，即从字符串的头部开始，有一个子串会一直重复出现。那么，这个子串的长度一定能整除字符串的长度，且每一个重复子串都是相同的，如何判断这么多的重复子串呢？利用动态规划的思想，从第二个重复子串开始，和第一个字符比较；第三个和第二个比较，一直到最后。这样，就能够判断完所有重复的子串。

1. 代码实现

Java：

class Solution {

    public boolean repeatedSubstringPattern(String s) {

        int n = s.length();

        for(int i = 1; i\*2 <= n; ++i){

            if(n % i == 0){

                boolean flag = true;

                for(int j = i; j  < n; ++j){

                    if(s.charAt(j) != s.charAt(j-i)){

                        flag = false;

                        break;

                    }

                }

                if(flag){

                    return true;

                }

            }

        }

        return false;

    }

}