1. 题目

（一）题干

实现 strStr() 函数。

给定一个 haystack 字符串和一个 needle 字符串，在 haystack 字符串中找出 needle 字符串出现的第一个位置 (从0开始)。如果不存在，则返回  -1。

（二）示例

示例 1:

输入: haystack = "hello", needle = "ll"

输出: 2

示例 2:

输入: haystack = "aaaaa", needle = "bba"

输出: -1

1. 题解
2. 思路

这是一个字符串匹配的问题，首先想到的当然是直接将needle字符串在haystack字符串上滑动，若在滑动窗口内的字符串是相同的，那么查找成功。

这种方法列举出了所有的可能性，是可以改进的，先在haystack字符串中查找出等于needle的第一个字符的字符位置，再对后面的字符进行循环，每次循环结束，若能够找到完全匹配的字符串，则返回该字符串初始位置；若不能，则回溯到本次循环的下一个位置处进行下一次循环。

1. 代码实现

Java:

初级思路：

class Solution {

public int strStr(String haystack, String needle) {

int L = needle.length(), n = haystack.length();

for (int start = 0; start < n - L + 1; ++start) {

if (haystack.substring(start, start + L).equals(needle)) {

return start;

}

}

return -1;

}

}

改进思路：

class Solution {

    public int strStr(String haystack, String needle) {

        int hl = haystack.length();

        int nl = needle.length();

        if(nl == 0){

            return 0;

        }

        int start = 0;

        while(start < hl - nl + 1){

            //从每个位置开始一次循环

            //找到第一个字符相同的位置

            while(start<hl-nl+1 && haystack.charAt(start)!=needle.charAt(0)){

                start++;

            }

            int curLength=0, nCur=0;

            //找匹配的字符串最大值

            while(start < hl && nCur < nl && haystack.charAt(start) == needle.charAt(nCur)){

                start++;

                nCur++;

                curLength++;

            }

            //判断curLength是否为nl

            //是，说明找到了，返回第一个字符位置

            if(curLength==nl){

                return start-nl;

            }

            //否则，回溯start

            start = start - curLength + 1;

        }

        return -1;

    }

}