# 题目

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。

**示例 1:**

**输入:** haystack = "hello", needle = "ll"

**输出:** 2

**示例 2:**

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

**输出:** -1

**说明:**

当 needle 是空字符串时，我们应当返回什么值呢？这是一个在面试中很好的问题。

对于本题而言，当needle是空字符串时我们应当返回0。这与C语言的strstr()以及Java的indexOf()定义相符。

# 分析

求子串问题：双指针+KMP。

## 方法一：双指针

class Solution {

public:

int strStr(string haystack, string needle) {

int i = 0;

int j = 0;

while(haystack[i]!='\0'&&needle[j]!='\0')

{

if(needle[j]==haystack[i])//判断是否相等

{

j++;

i++;

}

else//不相等退回开始的位置，i+1，j=0;

{

i = i - j + 1;

j = 0;

}

}

if(j == needle.length())//j为步长

return i-j;

return -1;

}

};

**复杂度分析：**

时间复杂度：最坏时间复杂度为O((N−L)L)，最优时间复杂度为O(N)。

空间复杂度：O(1)。

## 方法二：KMP

class Solution {

public:

int strStr(string haystack, string needle) {

if(!needle.size()) return 0;

if(!haystack.size()) return -1;

//先构造pattern

int j = -1, i = 0;//j在后面，i在前面

vector<int> b(needle.size() + 1);

b[i] = j;

while(i < needle.size())

{

while(j >= 0 && needle[i] != needle[j]) j = b[j];

i++, j++;

b[i] = j;

}

j = 0, i = 0; //j这回是text的， i是pattern的

while(j < haystack.size())

{

while(i >= 0 && needle[i] != haystack[j]) i = b[i];

i++, j++;

if(i == needle.size())

{

return j - needle.size();

}

}

return -1;

}

};