# 28、实现strStr()函数（Implement strStr()）

## 题目：

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 ndexOf()定义相符。

## 解答：

**法一：**

|  |
| --- |
| public static int StrStr(string haystack, string needle)  {  if (haystack == null || haystack.Length < needle.Length)  {  return -1;  }  if (needle == null || needle.Length == 0)  {  return 0;  }  //haystack.Length=5  for (int i = 0; i < haystack.Length; i++)  {  //len是目标字符串的长度  int len = 0;  //在寻找的位数少于原字符串，长度少于目标字符串，且原字符串的len+i处的字符和目标的一样则继续找下一位,没有就i下一个  //例如hello ll .i=0时hay的0位和need的0位不同，不循环i=1,这时hay的第1位和need0位不同，i=2，这时  //hay的2位和need的0位相同，进入循环len=1,这时hay的第3位和need的第1位相同，继续循环，len=2，这时hay第  //4位和need的第2位不同跳出循环，这时进入if判断  while (i + len < haystack.Length && len < needle.Length && haystack[len + i] == needle[len])  {  len++;  }  if (len == needle.Length)  {  return i;  }  }  return -1;  } |

**法二**：

|  |
| --- |
| public static int StrStr1(string haystack, string needle)  {  if (haystack == null || haystack.Length < needle.Length)  {  return -1;  }  if (needle == null || needle.Length == 0)  {  return 0;  }  int len = -1;  for (int i = 0; i < haystack.Length - needle.Length + 1; i++)  {  //这时i=2时满足  if (haystack[i] == needle[0])  {  if (IsStrStr(haystack, needle, i))  {  len = i;  break;  }  }  }  return len;  }  public static bool IsStrStr(string haystack, string needle, int k)  {  //k=2  bool flag = true;  //找到不同的点返回flase  for (int i = 0; i < needle.Length; i++)  {  if (needle[i] != haystack[i + k])  {  flag = false;  break;  }  }  return flag;  } |