

# 真实面试案例分析(下)

## 与 80% 求职者都会采坑的基础知识

一手微信 [soul382](#) 名企 offer 都有  
主讲人 令狐冲

# Implement strStr

一手微信study322 九章来offer都有

<http://www.lintcode.com/problem/strstr/>

在一个字符串中查询另外一个字符串第一次出现的位置

# 常见错误： 我知道一个算法叫做KMP

A同学: 论坛上有人说考到了KMP呢！你骗人！

```
1 class Solution:
2     """
3     @param source:
4     @param target:
5     @return: return the index
6     """
7     def strStr(self, source, target):
8         if not target:
9             return 0
10        for i in range(len(source)):
11            if source[i: i + len(target)] == target:
12                return i
13        return -1
```

```
1 public class Solution {
2     /**
3      * @param source:
4      * @param target:
5      * @return: return the index
6      */
7     public int strStr(String source, String target) {
8         if (target == null || target.equals("")) {
9             return 0;
10        }
11
12        for (int i = 0; i < source.length(); i++) {
13            if (source.substring(i, i + target.length()).equals(target))
14                return i;
15        }
16
17        return -1;
18    }
19 }
20 }
```

# $O(n^2)$ Best Practise

```
1 class Solution:
2     """
3     @param source:
4     @param target:
5     @return: return the index
6     """
7     def strStr(self, source, target):
8         if not target:
9             return 0
10
11        for i in range(len(source) - len(target) + 1):
12            for j in range(len(target)):
13                if source[i + j] != target[j]:
14                    break
15            else:
16                return i
17
18        return -1
```

```
1 public class Solution {
2     /**
3      * @param source:
4      * @param target:
5      * @return: return the index
6      */
7     public int strStr(String source, String target) {
8         if (target == null || target.equals("")) {
9             return 0;
10        }
11
12        for (int i = 0; i < source.length() - target.length() + 1; i++) {
13            boolean notEqual = false;
14            for (int j = 0; j < target.length(); j++) {
15                if (source.charAt(i + j) != target.charAt(j)) {
16                    notEqual = true;
17                    break;
18                }
19            }
20
21            if (!notEqual) {
22                return i;
23            }
24        }
25
26        return -1;
27    }
28 }
29 }
```

# 真问我比 $O(n^2)$ 更好的算法怎么办？

一手微信study322 九章来offer都有  
这个概率只有1%

可以学习一个比 KMP 算法更简单的算法: Rabin-Karp