

剑指Offer（二十七）：字符串的排列

🕒 2017年12月19日 10:19:12 🗨 4 🌡 6,391 °C 🖨 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

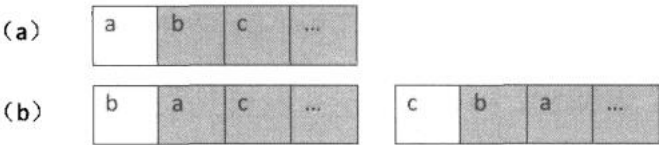
输入一个字符串,按字典序打印出该字符串中字符的所有排列。例如输入字符串abc, 则打印出由字符a,b,c所能排列出来的所有字符串abc,acb,bac,bca,cab和cba。

输入描述：

输入一个字符串,长度不超过9(可能有字符重复), 字符只包括大小写字母。

1、思路

我们求整个字符串的排列, 可以看成两步: 首先求所有可能出现在第一个位置的字符, 即把第一个字符和后面所有的字符交换。如下图所示:



上图就是分别把第一个字符a和后面的b、c等字符交换的情形。首先固定第一个字符, 求后面所有字符的排列。这个时候我们仍把后面的所有字符分为两部分: 后面的字符的第一个字符, 以及这个字符之后的所有字符。然后把第一个字符逐一和它后面的字符交换。

这个思路, 是典型的递归思路。

2、代码

C++:

```
1 class Solution {
2 public:
3     vector<string> Permutation(string str) {
4         //判断输入
5         if(str.length() == 0){
6             return result;
7         }
8         PermutationCore(str, 0);
9         //对结果进行排序
10        sort(result.begin(), result.end());
11        return result;
12    }
13
14 private:
15     void PermutationCore(string str, int begin){
16         //递归结束的条件: 第一位和最后一位交换完成
17     }
```


```
17     if(begin == str.length()){
18         result.push_back(str);
19         return;
20     }
21     for(int i = begin; i < str.length(); i++){
22         //如果字符串相同，则不交换
23         if(i != begin && str[i] == str[begin]){
24             continue;
25         }
26         //位置交换
27         swap(str[begin], str[i]);
28         //递归调用，前面begin+1的位置不变，后面的字符串全排列
29         PermutationCore(str, begin + 1);
30     }
31 }
32 vector<string> result;
33 };
```

Python:

	Pytho
<pre>1  #- coding:utf-8 -*- 2  class Solution: 3      def __init__(self): 4          self.result = [] 5      def Permutation(self, ss): 6          # write code here 7          if len(ss) == 0: 8              return [] 9          self.PermutationCore(ss, 0) 10         sorted(self.result) 11         return self.result 12     def PermutationCore(self, str_, begin): 13         if begin == len(str_): 14             self.result.append(str_) 15             return 16         for i in range(begin, len(str_)): 17             if i != begin and str_[i] == str_[begin]: 18                 continue 19             str_list = list(str_) 20             str_list[i], str_list[begin] = str_list[begin], str_list[i] 21             str_ = ''.join(str_list) 22             self.PermutationCore(str_, begin+1)</pre>	

Python:

	Pytho
<pre>1  #- coding:utf-8 -*- 2  class Solution: 3      def Permutation(self, ss): 4          if len(ss) &lt;=0: 5              return [] 6          res = list() 7          self.perm(ss,res,'') 8          uniq = list(set(res)) 9          return sorted(uniq) 10     def perm(self,ss,res,path): 11         if ss=='': 12             res.append(path) 13         else: 14             for i in range(len(ss)): 15                 self.perm(ss[:i]+ss[i+1:],res,path+ss[i])</pre>	



微信公众号

分享技术，乐享生活：微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

悲剧将人生的有价值的东西毁灭给人看，喜剧将那无价值的撕破给人看。--- 鲁迅