

## 个人资料



feliciafay



访问: 766348次

积分: 10344

等级: BLOG &gt; 7

排名: 第1012名

原创: 408篇 转载: 14篇

译文: 0篇 评论: 119条

## 文章搜索

## 博客专栏

1337Code  
C++

LeetCode(C++)

文章: 47篇  
阅读: 55994

## 文章分类

C++ (263)

Java (57)

DataStructure\_Algorithm (175)

DataMining\_MachineLearning (19)

InterviewQuestions (183)

Math (13)

Scripts (50)

Others (23)

System (1)

## 文章存档

2015年12月 (4)

2015年11月 (1)

2015年10月 (2)

2015年09月 (2)

[【CSDN会员专属福利】OpenStack Days China 大会门票, 先到先得](#) [【收藏】Html5 精品资源汇集](#) [我们为什么选择Java](#)

## LeetCode(22)GenerateParentheses

标签: LeetCode

2013-12-19 14:32

1138人阅读

评论(0)

收藏

举报

本文章已收录于:



算法与数据结构知识库

分类: C++ (262)

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

题目如下:

Given n pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

For example, given n = 3, a solution set is:

"((()))", "(()())", "(())()", "()(())", "()()()"

题目意思是说, 给出所有的合法的括号形式。比如, 手动地来写写。

n=1时, "()"

n=2时, "()", "()"

n=3时, "((()))", "(()())", "(())()", "()(())", "()()()"

这个问题比较适合用递归来解决。经过一些探索, 想到一个比较自然也比较好理解的办法。描述如下:

当n=1时, 显然只能写出"()"

当n=2时, 为什么写出了"()", "()", 这和n=1时候的结果有什么关系呢? 你可以看成有个基本元素element="()", 它尝试把它本身放入上一步的结果中, 也就是n=1的结果中。

当n=1时, 观察下面这个表达式

| ( | ) |

可以认为, 对上面表达式中的三个竖线而言, 每个竖线都是一个候选的插入位置。

如果把element="()" 插入竖线一, 将构成(())

如果把element="()" 插入竖线二, 将构成(())

如果把element="()" 插入竖线三, 将构成(())

其中过滤掉重复, 最后得到当n=2时, 合法的括号形式为"()", "()"

把上面的思考写成code, 如下

```
[cpp] view plain copy print ?  
01. //  
02. // Solution.h  
03. // LeetCode0J_021_GeneratePairsOfParenthese  
04. //  
05. // Created by feliciafay on 12/18/13.  
06. // Copyright (c) 2013 feliciafay. All rights reserved.  
07. //  
08. //  
09. #ifndef LeetCode0J_021_GeneratePairsOfParenthese_Solution_h  
10. #define LeetCode0J_021_GeneratePairsOfParenthese_Solution_h  
11. #include <vector>  
12. #include <iostream>  
13. #include <set>  
14. #include <string>  
15. #include <cmath>
```



收藏到代码笔记

2015年08月 (1)

展开

阅读排行

在ubuntu中配置SSH(解... (98485)  
使用svn diff的-r参数的来 (43763)  
二叉树系列(1)已知二叉树 (37271)  
为什么样本方差要除以n- (31123)  
看图理解单链表的反转 (13471)  
AWK使用总结(2)使用spli (12335)  
vector容器中存放结构体: (11598)  
LeetCode(5) Longest Pa (10509)  
马克(马克阅读C++的结 (8067)  
Xshell的session配置文件 (7997)

评论排行

为什么样本方差要除以n- (11)  
在ubuntu中配置SSH(解... (11)  
看图理解单链表的反转 (8)  
二叉树系列(1)已知二叉树 (8)  
再谈类的构造函数与初始 (6)  
LeetCode(5) Longest Pa (4)  
C++Primer学习笔记第二 (4)  
extern与static的链接属性 (4)  
LeetCode(139)Word Bre (4)  
LeetCode(53)Maximum (4)

最新评论

LeetCode(7) Reverse Integer  
Dark\_Alex1943: `` if(result > INT32\_MAX ) retu...  
LeetCode(115) Distinct Subseq...  
leonononon:  
bangbangbang!!!  
看图理解单链表的反转  
bacon0718: 方法3上面的代码有问题吧。第16行m->next=p->next; 第1次循环时, p->next指...  
LeetCode(53)Maximum Subarr...  
q714779883: 你这个记位置不对啊比如你这个start是1 end呢确是0 你这个不对  
LeetCode(137)Single NumberII  
葡萄架的程序猿:  
@chenchaofuck1:32\*n不是线性?  
CC150 Arrays and Strings 1.1 ~ qq\_34125070: 1.3 注释的题目内容和实现的代码好像不是一回事  
LeetCode(2) AddTwoNumbers  
Arthur\_Pendragon: 求问题解法一和解法二的区别==!  
LeetCode(53)Maximum Subarr...  
qq\_23732891: @qq\_23732891: 奥, 懂了  
LeetCode(53)Maximum Subarr...  
qq\_23732891: 疑惑,, , 最后一个算法如果是全负的情况怎么处理呢?  
LeetCode(53)Maximum Subarr...  
qq\_23732891: 漂亮啊

```
16. using namespace std;
17.
18. class Solution {
19. public:
20.     vector<string> generateParenthesis(int n) {
21.         if(n<1){
22.             vector<string> res_vec;
23.             return res_vec;
24.         }
25.         else if(n==1){
26.             vector<string> res_vec;
27.             res_vec.push_back("(");
28.             return res_vec;
29.         }else{
30.             vector<string> pre_vec=generateParenthesis(n-1);
31.             set<string> filter_set;
32.             vector<string> res_vec;
33.             for(int i=0;i<pre_vec.size();i++){
34.                 string pre_str=pre_vec[i];
35.                 int pre_str_len=(int)pre_str.length();
36.                 for(int j=0;j<pre_str_len;j++){
37.                     string now_str=pre_str.substr(0,j)+"(")+pre_str.substr(j,pre_s
38.
39.                     if(filter_set.find(now_str)==filter_set.end()){
40.                         filter_set.insert(now_str);
41.                         res_vec.push_back(now_str);
42.                     }
43.                 }
44.                 string now_str=pre_str+")";
45.                 if(filter_set.find(now_str)==filter_set.end()){
46.                     filter_set.insert(now_str);
47.                     res_vec.push_back(now_str);
48.                 }
49.             }
50.             return res_vec;
51.         }
52.     };
53.
54. #endif
```

小结

- (1) 一开始思考进入误区, 写了很久输出结果总是不全。后来换了角度看问题, 终于正确了。  
(2) 这个题目背后的奥秘是卡特兰数, 这个目前还没有开始看, 稍后更新。

顶 2  
踩 0

上一篇 [LeetCode\(20\) Valid Parentheses](#)  
下一篇 [LeetCode\(23\)Merge K Sorted Lists](#)

我的同类文章

C++ (262)

- LeetCode(140) Word Brea... 2015-04-10 阅读 558
- (zz) 趣题: 用最少的"并行... 2015-04-08 阅读 265
- LeetCode(198) Horse Ro... 2015-04-08 阅读 220
- LeetCode(126) Word Lad... 2015-04-07 阅读 586
- LeetCode(36) Valid Sudoku 2015-03-28 阅读 262
- LeetCode(174) Dungeon ... 2015-03-28 阅读 585
- Print A Tree In Vertical Ord... 2015-03-25 阅读 270
- LeetCode(188) Best Time... 2015-03-23 阅读 799
- LeetCode(37) Sudoku Sol... 2015-03-23 阅读 283
- LeetCode(190) Reverse B... 2015-03-22 阅读 1620
- LeetCode(76)Minimum W... 2015-03-22 阅读 355

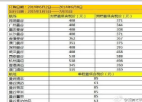
更多文章

## 猜你在找

|                                 |                                    |
|---------------------------------|------------------------------------|
| iOS8-Swift开发教程                  | leetcode 22 Generate Parentheses递归 |
| HTML 5移动开发从入门到精通                | leetcode22Generate Parentheses     |
| 老郭全套iOS开发课程【Objective-C】        | leetcode22 Generate Parentheses    |
| iOS移动开发从入门到精通 (Xcode7 & Swift2) | Leetcode22 Length of Last Word     |
| C++语言基础                         | LeetCode22Generate Parentheses     |



程序员招聘



泰国机票



哈尔滨到成都



卡特兰数



四川大学怎么



班级名字



班级的

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点,不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack  
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery  
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity  
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack  
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide  
Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase  
Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

