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:

[

"((()))",

"(()())",

"(())()",

"()(())",

"()()()"

]

这道题，就很麻烦。我是先找规律然后再做的。

规律是：首先你怎么变最前边和最后边的括号始终不变的，变的就是中间的括号。

然后找中间的规律，我们把（用1表示，）用2表示。

1122

1212

1221

2112

2121

除去第一个，然后就是剩下组合的排列组合。

说实话我从这道题里没有学会什么东西。答案也没怎么看懂，他的第一个答案是把所有的情况列出来然后找符合条件的，时间复杂度很高。第二种方法是错误的，可能他没写全。第三种方法是我的方法的简写，用递归写的，恕我无能确实看不懂也写不出来这样的代码。以后有机会回来看看。这里就不多说了。

**注意的是：以后要加强自己写递归算法的技能！！！**