Longest Valid Parentheses

Difficulty: Hard

这道题是要寻找给定的串s中最长的合法括号序列，给我的第一印象就感觉非常像最大连续子数组和。那下面也是类似的讨论3中复杂度的解法。

暴力

对每一个$i, 0 <= i < n$，考察以$s[i]$为起点的最长合法括号序列的长度。依次将$s[i]$及其之后的每一个元素入栈，遇到匹配就弹出。每次栈空的时候更新一下当前记录的最长的长度。

当然这种做法还是有常数上的优化空间的。假设已经直到以$s[i]$为起点的最长合法括号序列为k，那么以$s[i+1], s[i+2], \cdots, s[i+k-1]$为起点的最长合法括号序列的长度一定不会超过$k$，可以直接跳过。此时该算法在最好情况下可以达到$O(n)$的复杂度，但最坏情况还是$O(n^2)$.

分治

分的过程很简单，将串s分成两段，分别考察这两段的最长合法括号序列长度。

合的时候，从中间开始向两边遍历，用两个数c_1, c_2保存左右两边的匹配情况。向左边遍历时，遇到")"c_1加1，遇到"("$c1$减1，如果减1之前$c_1 == 0$，则遍历右半部分。遍历右半部分时的操作与左半部分完全对称。这样就得到了跨越中点的最长合法括号序列的长度。取这个值与左半部分，右半部分各自的最大值作为最终的结果。

时间复杂度的递推公式为

$$T(n) = 2T(n/2) + \Theta(n)$$

由Master Theory解得$T(n) = \Theta(n \lg n)$.

动态规划

同样，类似于暴力算法，我们计算以$s[i]$为起点的最长合法括号序列长度，记为$d[i]$.

显然，如果$s[i] ==$')'，那么$d[i] = 0$.

如果$s[i]$ =='(', 那么我们可以尝试寻找与其配对的')'。如果$s[i+1]$ ==')', 那么$s[i+1]$一定与$s[i]$ 匹配，此时$d[i] = 2 + d[i+2]$. 2表示$s[i]$与$s[i+1]$。如果$s[i+1] \neq$')'，记$c = d[i+1]$，那么与$s[i]$匹配的括号只有可能位于$i+c+1$的位置。如果$s[i+c+1] \neq$')'，那么$d[i] = 0$. 否则，
$d[i] = d[i+1] + 2 + d[i+c+2]$。 这个的意思是说，$s[i]$与$s[i+c+1]$匹配，二者中间有一个长度为$c$，也就是$d[i+1]$的合法括号序列。再加上以
$s[i+c+2]$为起点的最长合法括号序列长度，即为以$s[i]$为起点的最大长度。

写成递推公式为：

$$d[i] = \begin{cases} d[i+1] + 2 + d[i+c+2] & \text{if } s[i] == \text{'('} \text{ and } s[i+c+1] == \text{')'} \\ 0 & \text{otherwise} \end{cases}$$

其中$c = d[i+1]$.

最终数组d的最大值即为题目要求的最长合法括号序列的长度。算法复杂度$\Theta(n)$.