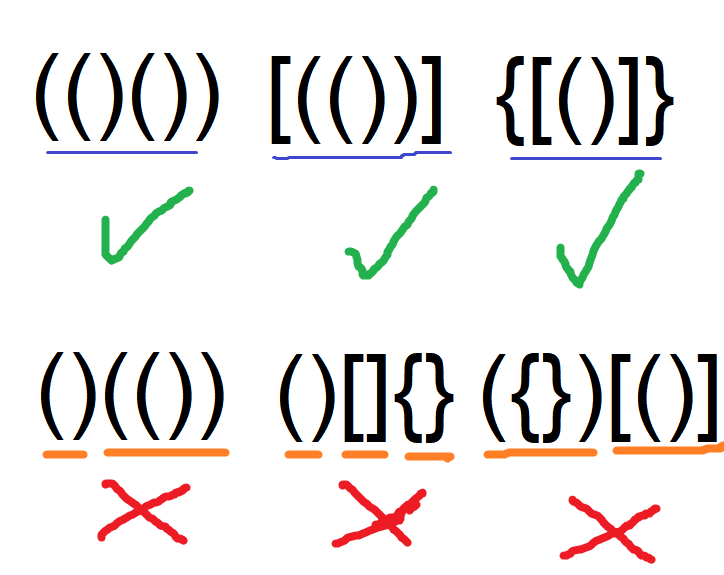
**Parenthesis 2**

เราจะมี dp สองตัวซึ่งจะนิยามดังนี้

- dp[0][l][r] = จำนวนวิธีในการสร้างวงเล็บสมดุลของตัวที่ l ถึงตัวที่ r โดยที่วงเล็บสมดุลที่ว่าต้องเป็นวงเล็บสมดุลพจน์เดี่ยวเท่านั้น(ดูรูปเพื่อความเข้าใจ)



- dp[1][l][r] = จำนวนวิธีในการสร้างวงเล็บสมดุลของตัวที่ l ถึงตัวที่ r

**Base case:**

1.เห็นได้ชัดว่า วงเล็บสมดุล(ไม่ว่าจะกรณีไหน)ต้องมีความยาวเป็นคู่เท่านั้น ถ้าความยาวคี่ จะไม่สามารถสร้างวงเล็บสมดุลได้แน่นอน นั่นคือ

If( (r-l+1)%2==1 ) return dp[0][l][r]=dp[1][l][r]=0;

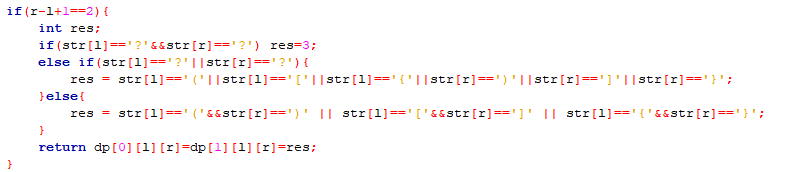
2.ถ้าเหลืออยู่ 2 ตัว ( r-l+1==2 )

- ถ้า str[l] และ str[r] คือ ‘?’ แสดงว่าเราจะกำหนดให้ str[l] กับ str[r] ได้ 3 วิธี คือ ‘(‘กับ’)’ หรือ ‘[‘กับ’]’ หรือ ‘{‘กับ’}’

- ถ้า str[l] หรือ str[r] คือ ‘?’ แสดงว่าถ้า str[l] เป็น ‘(‘หรือ’[‘หรือ’{‘ หรือ str[r] เป็น ‘)’หรือ’]’หรือ’}’ จะสร้างได้ 1 วิธี(คือให้คนที่เป็น ‘?’ เป็นตัวที่เข้าคู่กับอีกฝ่าย) นอกนั้นจะทำได้ 0 วิธี

- ถ้าไม่มีใครเป็น ‘?’ แสดงว่า str[l]กับstr[r] ต้องเป็นคู่วงเล็บเปิดปิดกัน คือ ‘(‘กับ’)’ หรือ ‘[‘กับ’]’ หรือ ‘{‘กับ’}’ จึงจะสร้างได้ 1 วิธี ถ้าไม่ใช่คู่วงเล็บเปิดปิดกันจะสร้างได้ 0 วิธี

สรุปคือ



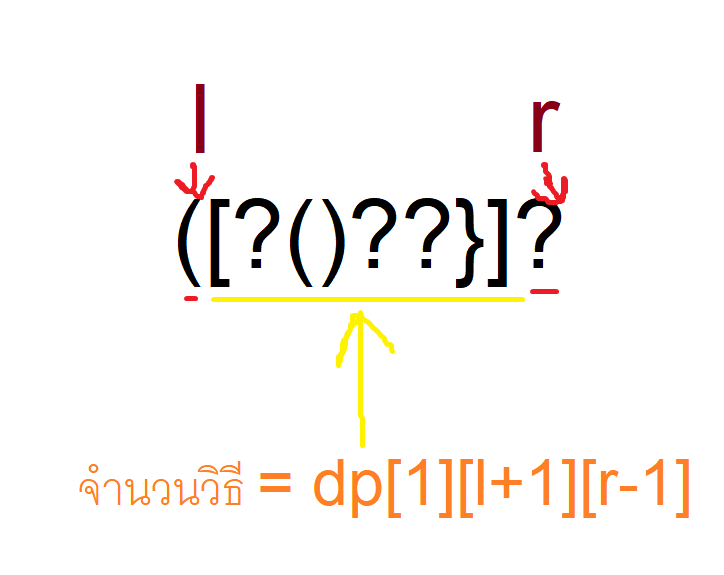
**จบ basecase**

ทีนี้ จากนิยาม เราลองมาดูวิธีคำนวณมิกทั้งสองแบบ

1. dp[0][l][r]

วิธีคำนวณจะไม่ยาก เนื่องจากเป็นจำนวนวิธีสร้างวงเล็บสมดุลพจน์เดี่ยวๆ ดังนั้นจำนวนวิธีจะเท่ากับ

(จำนวนวิธีที่ทำให้ตัวที่ l กับตัวที่ r เป็นวงเล็บเปิดปิดคู่กัน)xdp[1][l+1][r-1]



โดยเราจะสามารถแบ่งกรณีได้ดังนี้

**1. str[l]=str[r]=’?’**

สังเกตว่า เราจะสามารถให้ str[l]กับstr[r] เป็น ‘(‘กับ’)’ หรือ ‘[‘กับ’]’ หรือ ‘{‘กับ’}’ ซึ่งทำได้ 3 วิธี

**2. str[l]=’?’ หรือ str[r]=’?’ (กรณีในรูป)**

สังเกตว่า

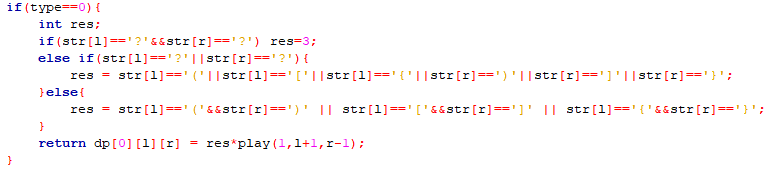
ถ้า str[l] = ‘?’ แล้ว str[r] ต้องเป็น ‘)’ หรือ ‘}’ หรือ ‘]’ เท่านั้น(ซึ่งทำได้ 1 วิธี คือให้ str[l] เป็นคู่ตรงข้ามกับ str[r]) ถ้าไม่ใช่ แสดงว่าทำได้ 0 วิธี

ถ้า str[r] = ‘?’ แล้ว str[l] ต้องเป็น ‘(’ หรือ ‘{’ หรือ ‘[’ เท่านั้น(ซึ่งทำได้ 1 วิธี คือให้ str[r] เป็นคู่ตรงข้ามกับ str[l]) ถ้าไม่ใช่ แสดงว่าทำได้ 0 วิธี

**3. ไม่มีใครเป็น ‘?’**

ในกรณีนี้ str[l] ต้องเป็น ‘(‘ หรือ ‘[‘ หรือ ‘{‘ เท่านั้น และ str[r] ก็ต้องเป็นคู่ตรงข้ามของ str[l] ด้วยเช่นกัน ถ้าเป็นจริงจะทำได้ 1 วิธี(ก็คือไม่ต้องกำหนดอะไรเลย) แต่ไม่จริงก็จะทำได้ 0 วิธี

สรุปก็คือ แทบไม่ต่างจาก base case แต่เราจะกำหนดค่าให้เฉพาะ dp[0][l][r]



ต่อมา

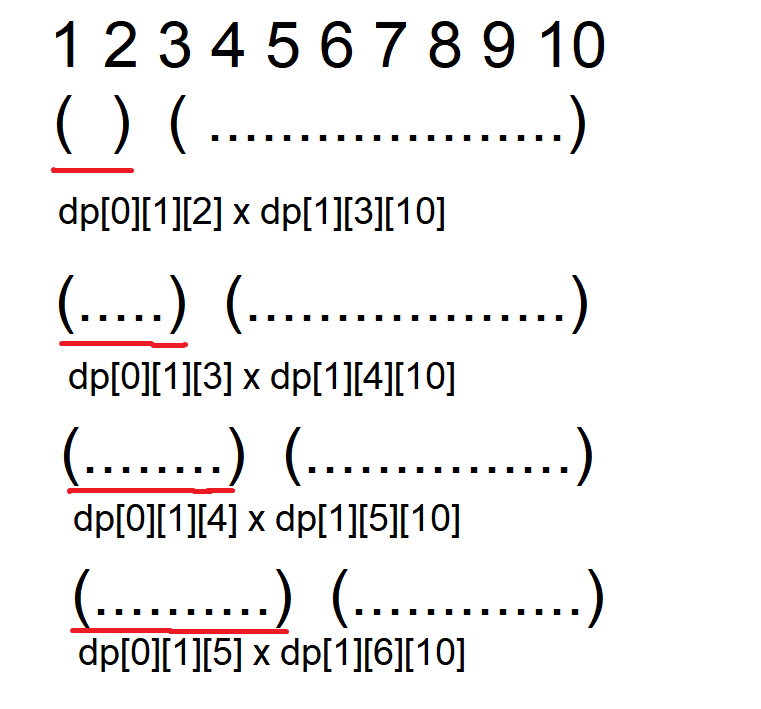
2. dp[1][l][r]

แน่นอนว่า มันต้องมีจำนวนวิธีของ dp[0][l][r] ด้วยแน่นอน

แต่ว่า dp[0][l][r] เป็นกรณีเฉพาะวงเล็บพจน์เดี่ยว

ดังนั้น เราจะมาดูกรณีหลายพจน์กัน

เราจะสนใจแค่วงเล็บสมดุลพจน์แรกเท่านั้น ส่วนพจน์หลังๆเราจะไปถาม dp[1][…][…] เอา(ดูรูปด้านล่าง)

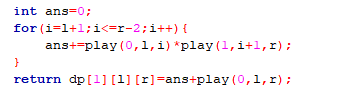


สังเกตว่า วิธีการคือ

สนใจแค่พจน์แรก แล้วส่วนที่อยู่หลังพจน์แรก ให้ไปถาม dp[1][…][r] เอา หรือก็คือ

dp[1][l][r] = dp[0][l][r] + for(l+1<=k<=r-2) dp[0][l][k]\*dp[k+1][r]

เขียนได้ดังนี้



**จบ**

ตอบก็ให้ตอบ dp[1][1][n] นั่นเอง