

一、試驗成果

今天有一根87及一根14單位長的rod，請大家利用老師上課所講的
top-down和**bottom-up**的方法找出要切割成幾根rod才能賣出多
少最高的價格，也請在報告中告訴我們最好的切法。

```
-----  
使用 top down 的方法 Cut rod  
總長度: 14  
價格: 40  
如何切: 2 2 10  
3根  
-----
```

```
-----  
使用 bottom up 的方法 Cut rod  
總長度: 14  
價格: 40  
如何切: 2 2 10  
3根  
-----
```

```
-----  
使用 top down 的方法 Cut rod  
總長度: 87  
價格: 258  
如何切: 1 6 10 10 10 10 10 10 10 10  
10根  
-----
```

```
-----  
使用 bottom up 的方法 Cut rod  
總長度: 87  
價格: 258  
如何切: 1 6 10 10 10 10 10 10 10 10  
10根  
-----
```

二、 討論

如果今天不是要找最高的價格的組合，而是最低的價格組合，你覺得演算法應該做怎樣的修正呢?(在結報中描述你的想法就可以了)

Ans: 很簡單，只要調整一些初始設定及後續的假設條件即可:

1) 初始設定

Memoized_Cut_Rod 函式裡面原本 r 設為除了第 0 項為 0，其餘 $1 \sim n$ 都是負無限大(INT_MIN)。改成除了第 0 項為 0，其餘 $1 \sim n$ 都是正無限大(INT_MAX)。

Memoized_Cut_Rod_Aux 函式裡面 \max 變數名稱改成叫做 \min ，初始值改成正無限大(INT_MAX)。Extended_Bottom_up_Cut_Rod 函式裡面 \max 變數名稱也改成叫做 \min 。

2) 假設條件

Memoized_Cut_Rod_Aux 函式裡面 $\text{if}(\max < d)$ 改成 $\text{if}(\min > d)$ 、

Extended_Bottom_up_Cut_Rod 函式裡面 $\text{if}(\max < p[i] + r[j-i])$ 及 $\text{else if}(\max < r[i] + r[j-i])$ 分別改成 $\text{if}(\min > p[i] + r[j-i])$ 及 $\text{else if}(\min > r[i] + r[j-i])$

三、 心得

這次作業實作動態規劃的經典 Cut rod 問題，我覺得很像玩益智遊戲，很需要動腦思考整個解法的設計。幸好講義上與網路上都有很多資源可以參考，若要從頭想演算法可能要花很多時間。不過這份作業出了一個講義沒有寫的問題，就是使用 top down 要怎麼記住每一刀切在哪裡。我參考了 bottom up 記住每一刀切在哪裡的方法，驚覺只要在 \max 要被更新的時候記住當下 for loop 執行到第幾個 iteration 就好了，因此我成功寫出來 top down 的部分。另外，我使用 top down 執行長度為 87 的 rod 會因為 recursive function 執行時間很長而跑不出來，我後來想到是因為 r 在 top down 時不能用 pass by value，會因為 function call 而刷新 r ，所以我就在 class 裡面新增一個 r ，這樣 r 就不會被洗掉。