思路紀錄

第2題  
先判斷哪個鍊錶長，然後用交換的方法確保一定是l1更長。  
然後把l2的值加到l1上，全部加完之後遍歷l1處理進位，處理最後一位需要進的特殊情況。

102

處理每一層的時候記錄下一層的節點，並把目前這一層每個節點的值記錄到結果裡。

103

二元樹的層序遍歷，只需要加一個變數lor（left or right）來判斷目前層是從左端開始，還是從右端開始。  
105  
首先根據前序遍歷的定義可以知道，preorder這個陣列的第一個元素preorder[0]一定是root，再根據中序遍歷的定義， 在inorder這個陣列裡，root前面的元素都屬於root的左子樹，root後面的元素都屬於右子樹，從這一步驟得到了left\_inorder和right\_inorder，把root在inorder裡的位置index = inorder.index(preorder[0])找出來，就可以知道其左子樹和右子樹的長度，然後再回到preorder，root後面先是左子樹，然後是右子樹，因為上一步我們已經知道了它們的長度，所以可以得到left\_preorder和left\_preorder。  
106  
跟上面差不多。  
107  
102的答案修改result为result[::-1]  
113  
找到每一条从根节点到叶节点的路径，然后判断路径之和是否等于需要的TARGET。卡了五分钟，因为把路径添加到res之后直接return了，而没有做POP的回溯操作。  
114  
開個陣列記錄前序遍歷，再一個個放上去  
120  
自底向上的动态规划。用dp[i][j] 表示从triangle[i][j]到最后一行的最小路径和。  
dp[i][j] = min(dp[i + 1][j], dp[i + 1][j + 1]) + triangle[i][j]。  
122  
要計算最大利潤，遵循貪心演算法的思想，也就是盡量在價格低的時候買入，在價格高的時候賣出。我們透過遍歷每天的價格，如果當天的價格比前一天的價格高，就計算一次利潤（即當天賣出前一天買入的股票的利潤），並將利潤累加。  
  
128 好難  
首先看題目要求O（n），用哈希表，建立一個哈希表，key是區間端點，val是這段區間的長度，線性掃描輸入數組，如果當前元素已經在雜湊表裡，則跳過，如果不在哈希表裡，就看一下它左右兩側區間的長度left， right，計算出它自身的區間長度length = 1 + left + right。計算完後更新新的左右端點的長度為，record[num - left] = length = record[num + right], record[num]也賦值，為了佔個位子。  
129  
先找到從根節點到葉節點的路徑，然後根據路徑計算出這一條路所代表的數字，最後把所有的數字加在一起。  
130  
從四邊上的O出發，用DFS把所有相鄰的O染成P。此時沒有被染色的點就是被圍繞的區域。再掃描一次，把所有的P變成O，把所有的O變成X。  
131  
給定一個字串 *s*，將 *s* 分割成一些子字串，使每個子字串都是[回文字串](https://so.csdn.net/so/search?q=%E5%9B%9E%E6%96%87%E4%B8%B2&spm=1001.2101.3001.7020)。傳回 *s* 所有可能的分割方案。  
133 難🥲  
利用一個哈希表mapping記錄所有的節點和它的copy的關係。  
137  
用哈希表記錄每個元素出現的頻率  
138  
開一個哈希表mapping，key是老結點，val是新節點，然後把老結點（key）的next和random對應的新節點（mapping[key.next] 和mapping[key.random]） 分別賦給新節點(val)的next （val.next）和random（val .random）。  
139  
動態規劃，用一個陣列record記錄切割字串s時下刀的下標。每次刷新最遠可以拆分的下標，判斷整個字串都可不可以被拆分。