```
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution(object):
    def addTwoNumbers(self, l1, l2):
        """
        :type l1: ListNode
        :type l2: ListNode
        :rtype: ListNode
        """

        if self.getLength(l1) < self.getLength(l2): #如果L1小於L2，就讓他們交換
            l1,l2=l2,l1

        head=l1
        while(l2):
            l1.val+=l2.val (把 l2 + 到 l1)
            l1=l1.next
            l2=l2.next

        p=head (處理進位)
        while(p):
            if p.val>9: (if 值 >9，進位)
                p.val-=10
                if p.next:
                    p.next.val+=1
                else:
                    p.next=ListNode(1)
            p=p.next
        return head

    def getLength(self,l):
        length=0
        while(l):
            length+=1
            l=l.next
        return length
```

定義.

```python
class Solution(object):
    def lengthOfLongestSubstring(self, s):
        """
        :type s: str
        :rtype: int
        """
        record = dict()
        res,start=0,0
        for end in range(len(s)):
            if s[end] in record:
                start=max(start,record[s[end]]+1)
            record[s[end]]=end
            res=max(res,end-start+1)
        return res
```

字典.

設都0.

檢查 dict 裡有無重複的.

前不穩

renew dict