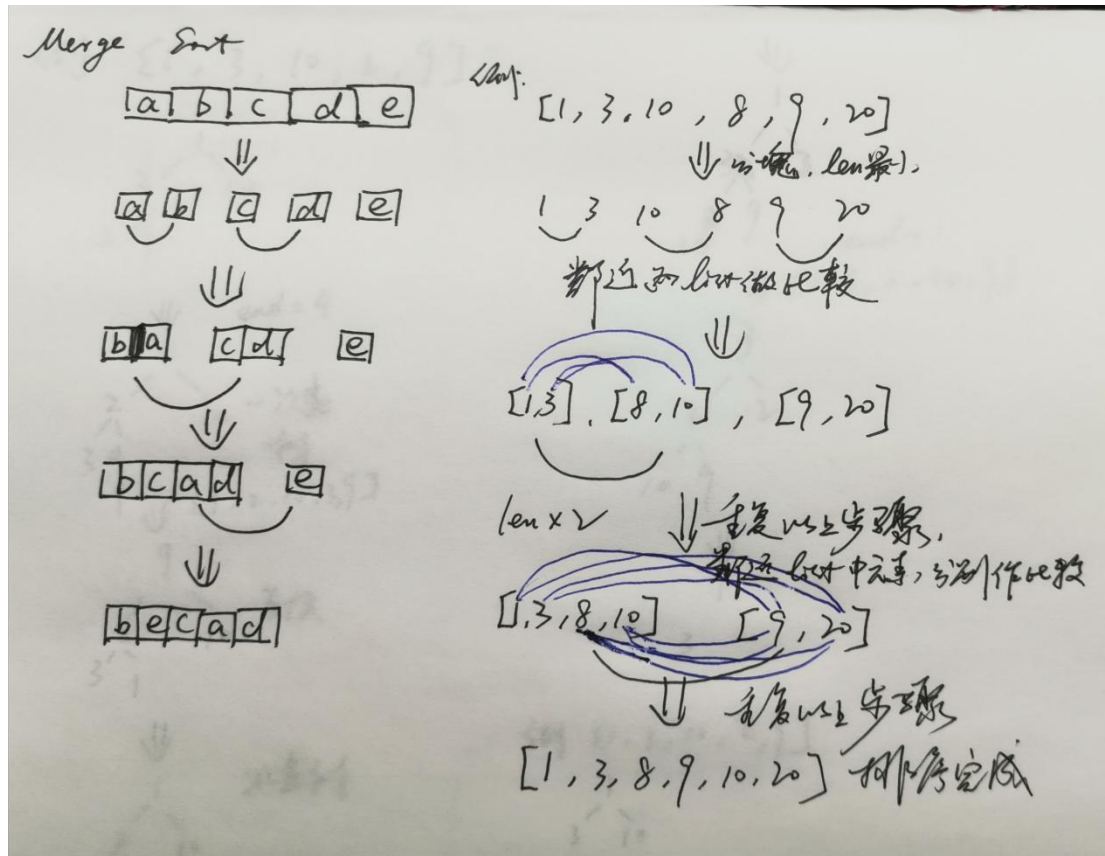


# Merge sort

\*註：引用資料採用藍色，我的說明採用紅色

## 1、流程圖



## 2、學習歷程

参考了 <https://baike.baidu.com/item/归并排序/1639015?fr=aladdin> 中的归并操作:

歸併操作(merge)，也叫歸併演算法，指的是將兩個順序序列合併成一個順序序列的方法。

如 設有數列{6, 202, 100, 301, 38, 8, 1}

初始狀態: 6,202,100,301,38,8,1

第一次歸併後: {6,202},{100,301},{8,38},{1}, 比較次數: 3;

第二次歸併後: {6,100,202,301}, {1,8,38}, 比較次數: 4;

第三次歸併後：{1,6,8,38,100,202,301},比較次數：4;

總的比較次數為： $3+4+4=11$ ;

逆序數為 14;

### 及演算法描述

歸併操作的工作原理如下：

第一步：申請空間，使其大小為兩個已經排序序列之和，該空間用來存放合併後的序列

第二步：設定兩個指針，最初位置分別為兩個已經排序序列的起始位置

第三步：比較兩個指針所指向的元素，選擇相對小的元素放入到合併空間，並移動指針到下一位置

重複步驟 3 直到某一指針超出序列尾

將另一序列剩下的所有元素直接複製到合併序列尾

及以下代碼：

```
Python语言

def MergeSort(lists):
    if len(lists) <= 1:
        return lists
    num = int( len(lists) / 2 )
    left = MergeSort(lists[:num])
    right = MergeSort(lists[num:])
    return Merge(left, right)
def Merge(left,right):
    r, l=0, 0
    result=[]
    while l<len(left) and r<len(right):
        if left[l] <= right[r]:
            result.append(left[l])
            l += 1
        else:
            result.append(right[r])
            r += 1
    result += list(left[l:])
    result += list(right[r:])
    return result
print MergeSort([1, 2, 3, 4, 5, 6, 7, 90, 21, 23, 45])
```

[【參考代碼】](#)

```
left = MergeSort(lists[:num])
right = MergeSort(lists[num:])
return Merge(left, right)
```

其中這段看不是很懂，但知道是用來切割的，所以做了如下改動

```
left=list[:n]
right=list[n:]
return merge(mergesort(left),mergesort(right)) #分別再對左右兩塊進行各自的左右分塊
```

Def merge 的部分本想做些改動，  
比如：

```
In [13]: def mergesort(list):
          if len(list)<=1:
              return list
          n=int(len(list)/2) #分為左右鄰近的兩塊
          left=list[:n]
          right=list[n:]
          return merge(mergesort(left),mergesort(right)) #分別再對左右兩塊進行各自的左右分塊

          def merge(left, right):
              n,m=0, 0
              result=[]
              for l in range(len(left)-1): #left中的數分別與right中的數進行比較、插入
                  for r in range(len(right)-1):
                      if left[l]<=right[r]: #由於list內部已是比過大小的，所以若left第一個值小於right第一個值，則必定小於right中其他值，則無需再比較
                          result.append(left[l])
                          n+=1
                          break
                      else:
                          result.append(right[r])
                          m+=1
              result+=list(left[n:])
              result+=list(right[m:]) #將剩餘的數加回結果list後，防止值的丟失

          return result
```

但出現如下錯誤：

```

TypeError                                Traceback (most recent call last)
<ipython-input-36-ed74e471992e> in <module>
      1 list1=[2,10,3,6,0,15]
----> 2 mergesort(list1)

<ipython-input-35-dfb9df4ed2ba> in mergesort(list)
      5     left=list[:n]
      6     right=list[n:]
----> 7     return merge(mergesort(left),mergesort(right))#分別再對左右兩塊進行各自的左右分塊
      8
      9

<ipython-input-35-dfb9df4ed2ba> in mergesort(list)
      5     left=list[:n]
      6     right=list[n:]
----> 7     return merge(mergesort(left),mergesort(right))#分別再對左右兩塊進行各自的左右分塊
      8
      9

<ipython-input-35-dfb9df4ed2ba> in mergesort(list)
      5     left=list[:n]
      6     right=list[n:]
----> 7     return merge(mergesort(left),mergesort(right))#分別再對左右兩塊進行各自的左右分塊
      8
      9

<ipython-input-35-dfb9df4ed2ba> in merge(left, right)
     11     n,m=0,0
     12     result=[]
----> 13     for l in (len(left)-1):#left中的數分別與right中的數進行比較、插入
     14         for r in (len(right)-1):
     15             if left[l]<right[r]:#由於list內部已是比過大小的，所以若left第一個值小於right第一個值，則必定小於right中其他值，則無需
再比較

TypeError: 'int' object is not iterable

```

以及將原本的將小的數值逐個放入，改為將大的值逐個放入：

```

def mergesort(list):
    if len(list)<=1:
        return list
    n=int(len(list)/2)#分為左右臨近的兩塊
    left=list[:n]
    right=list[n:]
    return merge(mergesort(left),mergesort(right))#分別再對左右兩塊進行各自的左右分塊

def merge(left,right):
    l=len(left)-1
    r=len(right)-1
    result=[]
    while l>=0 and r>=0:#left中的數分別與right中的數進行比較、插入
        if left[l]>right[r]:#由於list內部已是比過大小的，所以若left第一個值小於right第一個值，則必定小於right中其他值，則無需再比較
            result.insert(0,left[l])
            l-=1
        else:
            result.insert(0,right[r])
            r-=1
    result.insert(0,list(right[:r]))
    result.insert(0,list(left[:l]))#將剩餘的數加回結果list前，防止值的丟失

    return result

```

但出現如下錯誤：

```

TypeError                                Traceback (most recent call last)
<ipython-input-38-ed74e471992e> in <module>
      1 list1=[2,10,3,6,0,15]
----> 2 mergesort(list1)

<ipython-input-37-f7fda3aa7a60> in mergesort(list)
      5     left=list[:n]
      6     right=list[n:]
----> 7     return merge(mergesort(left),mergesort(right))#分別再對左右兩塊進行各自的左右分塊
      8
      9

<ipython-input-37-f7fda3aa7a60> in mergesort(list)
      5     left=list[:n]
      6     right=list[n:]
----> 7     return merge(mergesort(left),mergesort(right))#分別再對左右兩塊進行各自的左右分塊
      8
      9

<ipython-input-37-f7fda3aa7a60> in merge(left, right)
     13     result=[]
     14     while l>=0 and r>=0:#left中的數分別與right中的數進行比較、插入
----> 15         if left[l]>=right[r]:#由於list內部已是比過大小的，所以若left第一個值小於right第一個值，則必定小於right中其他值，則無需再
比較
     16             result.insert(0,left[l])
     17             l-=1

TypeError: '>=' not supported between instances of 'int' and 'list'

```

### 3、文字說明

```
In [11]: def mergesort(list):
        if len(list)<=1:
            return list
        n=int(len(list)/2) #分為左右鄰近的兩塊
        left=list[:n]
        right=list[n:]
        return merge(mergesort(left),mergesort(right)) #分別再對左右兩塊進行各自的左右分塊

        def merge(left,right):
            r,l=0,0
            result=[]
            while l<len(left) and r<len(right): #left中的數分別與right中的數進行比較、插入
                if left[l]<=right[r]: #由於list內部已是比過大小的，所以若left第一個值小於right第一個值，則必定小於right中其他值，則無需再比較
                    result.append(left[l])
                    l+=1
                else:
                    result.append(right[r])
                    r+=1
            result+=list(left[l:])
            result+=list(right[r:]) #將剩餘的數加回結果list後，防止值的丟失

        return result
```

merge sort 之原理為先將數列切割為最小的 list（長度為 1）後比較組成長度為二倍的 list，再將鄰組之間的數值分別比較插入；如此循環，直到 list 的長度變為原 list 的長度為止

以下為測試1:

```
In [40]: list1=[2,10,3,6,0,15]
        mergesort(list1)

Out[40]: [0, 2, 3, 6, 10, 15]
```

以下為測試2:

```
In [13]: list2=[3,5,-1,10,3,100,50,30,60,70,25,36,0]
        mergesort(list2)

Out[13]: [-1, 0, 3, 3, 5, 10, 25, 30, 36, 50, 60, 70, 100]
```

#### 4、參考資料

<https://baike.baidu.com/item/归并排序/1639015?fr=aladdin>