In [10]:

#合併排序法

合併排序主要區分為拆分與合併

拆分

1. 把大陣列切一半成為兩個小陣列
2. 把切好的兩個小陣列再各自切一半
3. 重複步驟二直到每個小陣列都只剩一個元素

合併

1. 排序兩個只剩一個元素的小陣列並合併
2. 把兩邊排序好的小陣列合併並排序成一個陣列
3. 重複步驟二直到所有小陣列都合併成一個大陣列

```
---------------------------------------------------------
------------------
NameError                                 Traceback (most recen
t call last)
<ipython-input-10-9395c2a6bc8b> in <module>
     47
     48 #Output
---> 49 output = Solution().merge_sort([2,5,6,8,7,9])
     50 output

<ipython-input-10-9395c2a6bc8b> in merge_sort(self, nums)
     10         self.divide_sort(num1, len(num1)-1)
     11         self.divide_sort(num2, len(num2)-1)
---> 12          self.combine(len(num1),len(num2))
     13
     14         return nums

<ipython-input-10-9395c2a6bc8b> in combine(self, size1, si
ze2)
     32         arg1=0
     33         arg2=0
---> 34          num1.append(999)
     35         num2.append(999)
     36

NameError: name 'num1' is not defined
```

```
class Solution(object):

    #MergeSort
    def merge_sort(self, nums):
        self.nums = nums
        num1=nums[:len(nums)//2]
        num2=nums[len(nums)//2:]
        num3=[]

        self.divide_sort(num1, len(num1)-1)
        self.divide_sort(num2, len(num2)-1)
        self.combine(len(num1),len(num2))

        return nums

    #Divide
    def divide_sort(self,num,size):
        for i in range(size-1):
            for base in range(size-1-i):
                min=base
                for compare in range(size):
                    compare=base+1
                    if num[compare]<num[min]:
                        min=compare

                    temp=num[min]
                    num[min]=num[base]
                    num[base]=temp
#Combine
    def combine(self,size1,size2):
        arg1=0
        arg2=0
        num1.append(999)
        num2.append(999)

        for arg3 in range(size1+size2):
            if num1[arg1]<num2[arg2]:
                num3.append(num1[arg1])
```

```
            arg1+=1
        else:
            num3.append(num2[arg2])
            arg2+=1




#Output
output = Solution().merge_sort([2,5,6,8,7,9])
output
class Sol(object):
    def try1(self, ar):
        self.ar=ar
        ar1 = ar[:len(ar)//2]
        ar2 = ar[len(ar)//2:]

        self.try2(ar1, len(ar1))

        return ar1

    def try2(self, ar, siz):
        k=2
        for k in range(siz):
            ar[k] = ar[k]+1



output=Sol().try1([1,2,3,4,5,6])
output
arr=[2,9,8,4,5]
siz=len(arr)
for i in range(siz-1):
    for i in range(siz-1-i):
        m=i
        for j in range(siz):
            j=i+1

            if arr[j]<arr[m]:
```

```
            m=j

        t=arr[m]
        arr[m]=arr[i]
        arr[i]=t



[2, 4, 5, 8, 9]

print(arr)
arr1=[2,4,6,8]
arr2=[1,3,5,7]
arr3=[]

siz1=len(arr1)
siz2=len(arr2)

index1=0
index2=0
arr1.append(999)
arr2.append(999)

for index3 in range(siz1+siz2):
    if arr1[index1]<arr2[index2]:
        arr3.append(arr1[index1])
        index1+=1
    else:
        arr3.append(arr2[index2])
        index2+=1

print(arr3)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

參考資料 https://medium.com/appworks-school/%E5%88%9D%E5%AD%B8%E8%80%85%E5%AD%B8%E6%BC%94%E7%AE%97%E6%B3%95-%E6%8E%92%E5%BA%8F%E6%B3%95%E9%80%B2%E9%9A%8E-%E5%90%88%E4%BD%B5%E6%8E%92%E5%BA%8F%E6%B3%95-6252651c6f7e