

# Merge Sort

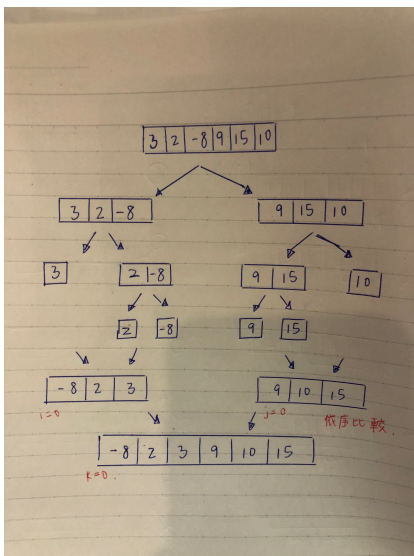
## 1.問題闡述

完成Merge Sort程式碼，老師在上課時就有在黑板解說Merge Sort的規則，也有請同學上去寫步驟，清晰許多，了解到Merge Sort就是把一個Array分割再分割接著再合併，並不是太難理解。

## 2.程式架構設計說明

依照助教給得作業格式，我先建立一個class類別第一個函式分割這個數列，這邊我有特別用print("nums" + str(nums))讓我清楚看到我這串數列分割下來的過程，並註記如果這個數列小於等於1，就return array。分割不能再分割後就return merge(left, right)，第二步就來定義merge(left, right)，創建新[]，左右去比較把小的數，依序放進新[]。

## 3.流程圖



給予一串數列，把它做分割分到不能再分的時候，要整合分割後的數列，這時要做合併的動作，合併時要將數列由小要大整合至兩大群組。

左右排序好的群組在進行合併，比較後把較小的數放下去，依序排列，直到完成。

## 4.逐步解釋說明

建立類別class，define裡頭的函式，依照作業格式的規定參數必須是self,nums。首先我要依序看到分割的情形我print("nums" + str(nums))必須要加str，因為要為字串格式，如果我今天的數列小於等於1，必須要return nums，當作遞回終止條件，找出midpoint作為分割，left,right訂出分割的規則，分到不能再分的時候，就return他到merge函式，merge函示他會把 left 跟 right 合併，而merge\_sort是把東西分割。

```
In [14]: class Solution(object):

    def merge_sort(self,nums):
        print("nums"+ str(nums))
        if len(nums) <= 1:
            return nums

        midpoint = int(len(nums) / 2)

        left = self.merge_sort(nums[:midpoint])
        right = self.merge_sort(nums[midpoint:])
        return self.merge(left, right)
```

接下來的merge函式是要來規定排列的問題，建立[]，左右群組index為0開始，就是向老師上可在黑板演練的，再來while那行因為是兩個陣列，所以要確保他不會超出界，小於的話就append進去[]，這時小於的數的群組就會往前加1，回傳result。

```
def merge(self, m_left, m_right):
    result = []
    left_pointer = right_pointer = 0
    while left_pointer < len(m_left) and right_pointer < len(m_right):
        if m_left[left_pointer] < m_right[right_pointer]:
            result.append(m_left[left_pointer])
            left_pointer += 1
        else:
            result.append(m_right[right_pointer])
            right_pointer += 1

    result.extend(m_left[left_pointer:])
    result.extend(m_right[right_pointer:])

    return result
```

## 5.測試值使用用範例例

```
output = Solution().merge_sort([3,2,-4,6,4,2,19])
print (output)
```

```
nums[3, 2, -4, 6, 4, 2, 19]
nums[3, 2, -4]
nums[3]
nums[2, -4]
nums[2]
nums[-4]
nums[6, 4, 2, 19]
nums[6, 4]
nums[6]
nums[4]
nums[2, 19]
nums[2]
nums[19]
[-4, 2, 2, 3, 4, 6, 19]
```

```
output = Solution().merge_sort([4,23,-5,17,7,14])
print (output)
```

```
nums[4, 23, -5, 17, 7, 14]
nums[4, 23, -5]
nums[4]
nums[23, -5]
nums[23]
nums[-5]
nums[17, 7, 14]
nums[17]
nums[7, 14]
nums[7]
nums[14]
[-5, 4, 7, 14, 17, 23]
```

```
output = Solution().merge_sort([54,-19,6,12,10,6,1])
print (output)
```

```
nums[54, -19, 6, 12, 10, 6, 1]
nums[54, -19, 6]
nums[54]
nums[-19, 6]
nums[-19]
nums[6]
nums[12, 10, 6, 1]
nums[12, 10]
nums[12]
nums[10]
nums[6, 1]
nums[6]
nums[1]
[-19, 1, 6, 6, 10, 12, 54]
```

## 6.其他補充說明

- Append() and extend() in Python

Append () : 將其參數作為單個元素添加到列表的末尾，列表的長度增加一，列表是一個對象，如果將另一個列表追加到列表中，則第一個列表將是列表末尾的單個對象。

extend () : 遍歷其參數，並將每個元素添加到列表中並擴展列表，列表的長度根據其參數中元素的數量而增加。

## 7.學習歷程

```
        r = + 1

        alist.extend(left[leftp:])
        alist.extend(right[rightq:])

        return list

output=Solution().merge_sort([3,2,-4,6,4,2,19])
print(output)

-----
NameError                                Traceback (most recent call last)
<ipython-input-28-0bcc5885f32a> in <module>()
     32
     33
--> 34 output=Solution().merge_sort([3,2,-4,6,4,2,19])
     35 print(output)

<ipython-input-28-0bcc5885f32a> in merge_sort(self, list)
      8         right = list[pivot:]
      9
--> 10         return merge_sort(left,right)
     11
     12         mergesort(left)

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

```
TypeError                                Traceback (most recent call last)
<ipython-input-1-d1f1fc862d86> in <module>()
     37
     38 if __name__ == "__main__":
--> 39     main()

<ipython-input-1-d1f1fc862d86> in main()
     33     print(list)
     34
--> 35     array = mergesort(list)
     36     print(array)
     37

<ipython-input-1-d1f1fc862d86> in mergesort(list)
      4
      5     pivot = len(list)//2
--> 6     left = mergesort(list[:pivot])
      7     right = mergesort(list[pivot:])
      8

<ipython-input-1-d1f1fc862d86> in mergesort(list)
      7     right = mergesort(list[pivot:])
      8
--> 9     return merge(left,right)
     10
     11 def merge(left,right):

<ipython-input-1-d1f1fc862d86> in merge(left, right)
     14     leftp = rightq = 0
     15
--> 16     while leftp < len[left] and rightq < len[right]:
     17
     18         if leftp < rightq :
```

```
TypeError: 'builtin_function_or_method' object is not subscriptable
```

過程中，遇到最大的困難就是我的函式沒有被定義到，“name” is not defined，意思是變量或者函數名拼寫錯誤或是不認識他，重新檢查注意縮排。

## 8. 參考資料

- <https://www.geeksforgeeks.org/merge-sort/>
- <https://www.youtube.com/watch?v=zfEmcm-eRbE&t=73s>
- [https://www.youtube.com/watch?v=CB\\_NCoxzQnk](https://www.youtube.com/watch?v=CB_NCoxzQnk)
- <http://notepad.yehyeh.net/Content/Algorithm/Sort/Merge/Merge.php>
- <https://kopu.chat/2017/08/10/合併排序-merge-sort/>
- <https://www.pyscoop.com/merge-sort-in-python/>