



國立中山大學資訊管理所

資料結構 上機實習

助教：林妤姍、陳亞琦、謝博丞
指導教授：李偉柏 博士

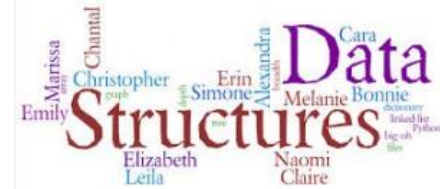


圖 (Graph)

- 一種有向或無向的資料結構
- 由數條邊 (Edge) 跟數個點 (Vertex) 組成
- 圖的相關應用
 - 最短路徑與成本 Shortest paths

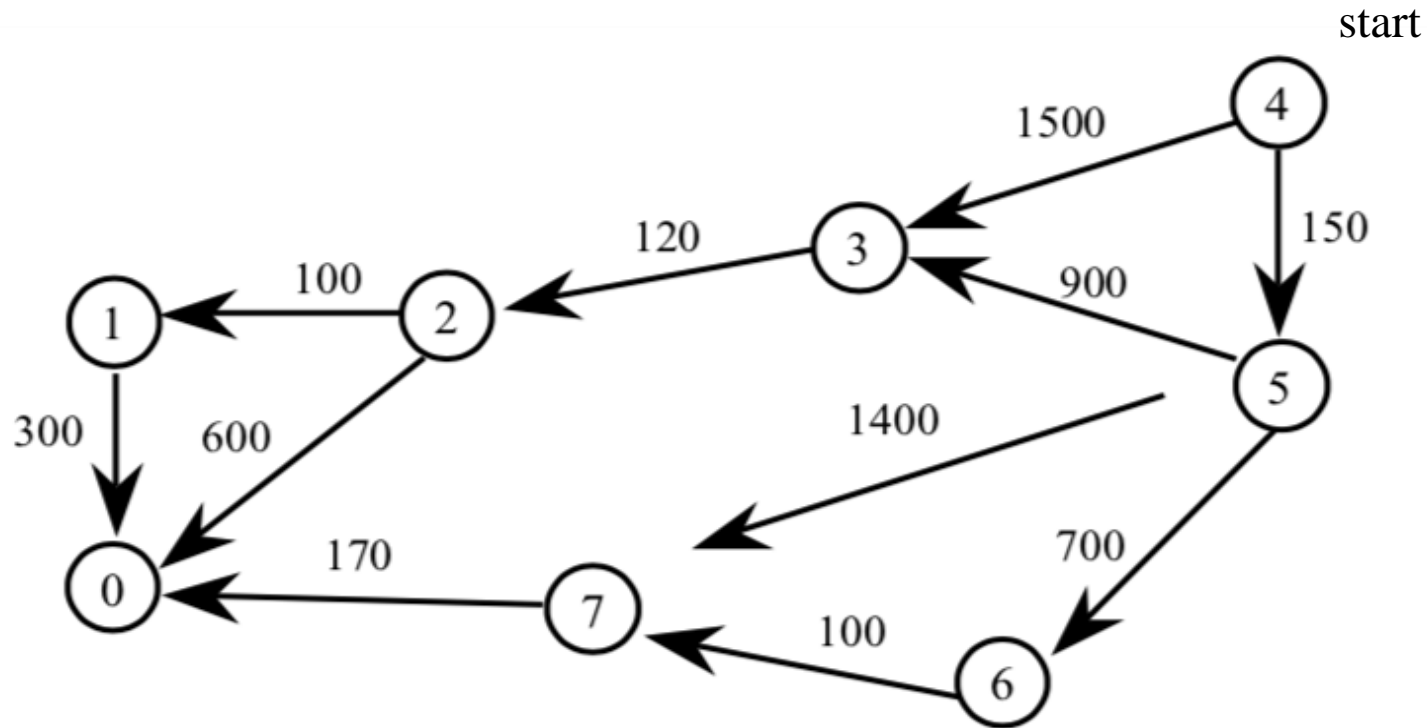


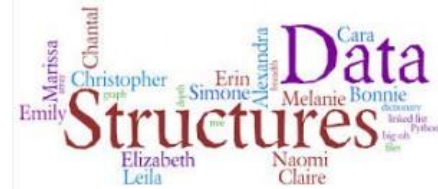


圖 (Graph)



- Single Source All Destinations





上機練習(一)

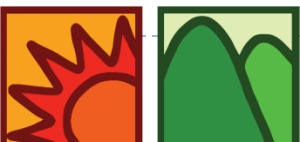
- 題目已給定圖跟最短路徑距離，完成以下問題：

1. 列出起點到各點的最短路徑
2. 在最短路徑後方附上對應的距離

- 參考答案

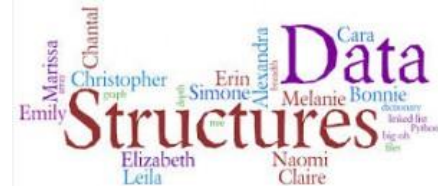
```
4 to all shortest distances:1120 1270 1170 1050 0 150 850 950
0 <- 7 <- 6 <- 5 <- 4 1120
1 <- 2 <- 3 <- 5 <- 4 1270
2 <- 3 <- 5 <- 4 1170
3 <- 5 <- 4 1050
4 <- 4 0
5 <- 4 150
6 <- 5 <- 4 850
7 <- 6 <- 5 <- 4 950
```

- 作業程式：[hw]shortest_path.c

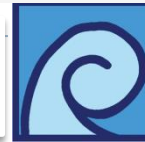
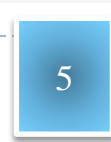
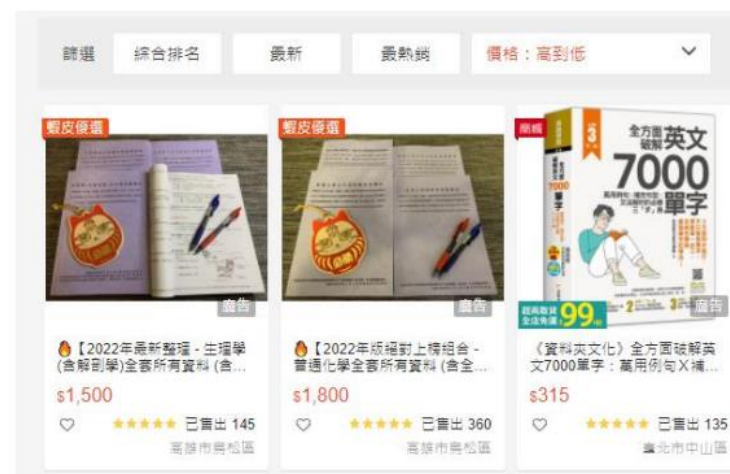


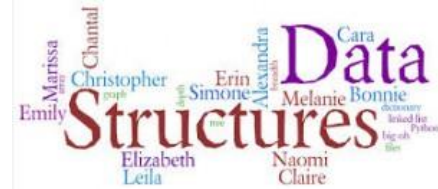


排序(sorting)



- 一種在資料處理上非常實用的方法
- 相關應用
 - 玩撲克牌、打麻將、金錢分類、線上購物
- 是許多資料結構跟演算法的先行處理方法
 - 統計資料中位數、最大最小值、最小生成樹、距離與成本排序





合併排序法 (Merge Sort)

- 屬於 Divide and Conquer 演算法
- 透過逐一排序子陣列後，合併排序結果來完成排序問題
- 拆分(Divide)
 1. 把待排序的陣列對半切分成兩個小陣列
 2. 把兩個小陣列再對半切分
 3. 重複步驟二，直到每個小陣列都只剩下一個元素
- 合併(Conquer)
 1. 排序兩個只剩一個元素的小陣列並合併
 2. 把左右兩邊排序好的小陣列合併並排序
 3. 重複步驟二，直到每個小陣列都被合併完成

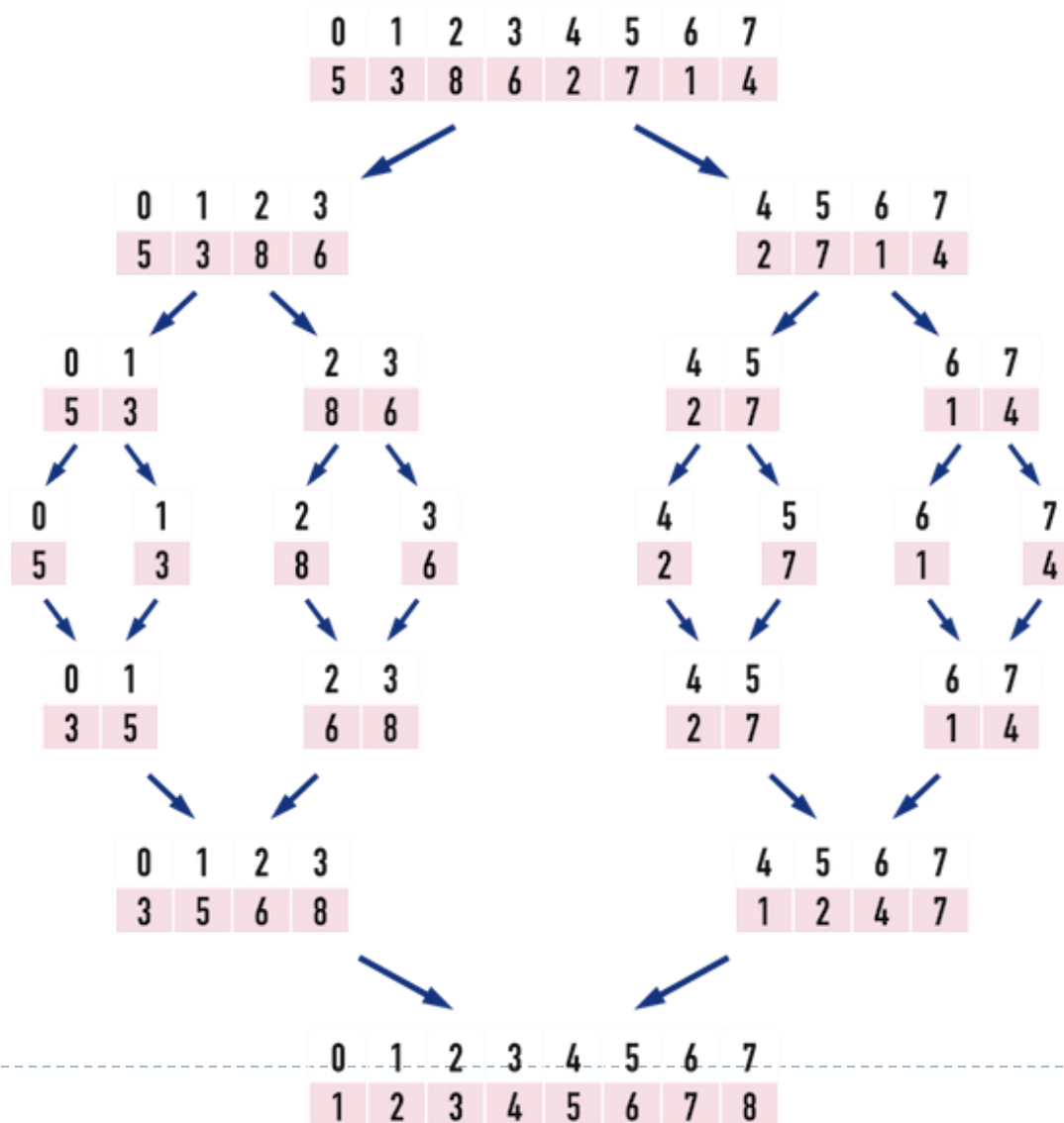


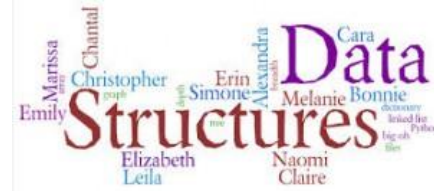


合併排序法 (Merge Sort)

Merge Sort

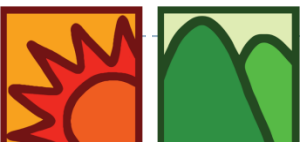
Divide
&
Conquer

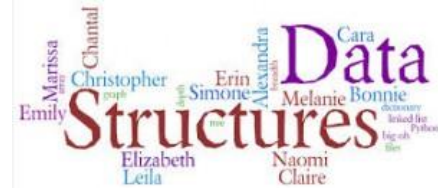




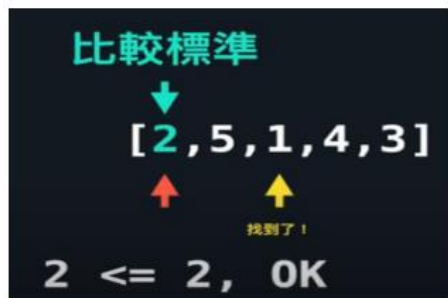
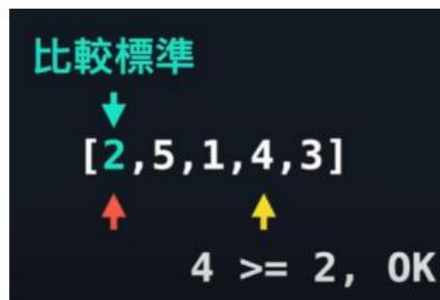
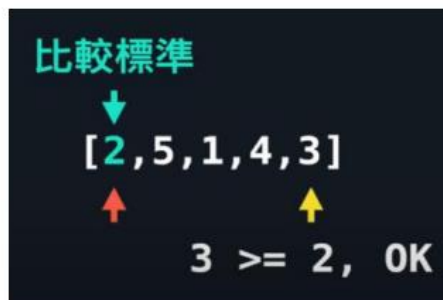
快速排序法 (Quick Sort)

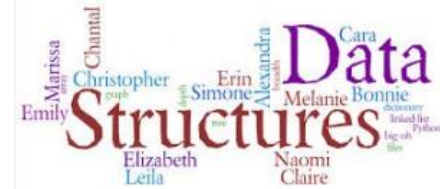
- 屬於 Divide and Conquer 演算法
- 在陣列中任意挑選一個數 (pivot)，然後調整陣列使其滿足以下條件：
 - 所有在 pivot 左邊的數都比 pivot 小
 - 所有在 pivot 右邊的數都比 pivot 大
- 循環步驟
 - 將所有在 pivot 左邊的所有數跟右邊的所有數視為兩個新的陣列
 - 使得新陣列滿足上述條件，直到分不出新的陣列為止





快速排序法 (Quick Sort)



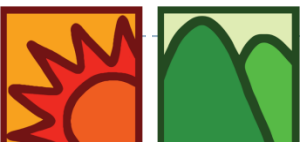


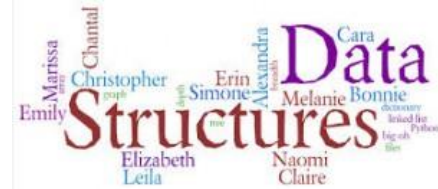
上機練習(二)

- 題目給定一個陣列，完成以下問題：
 1. 使用合併排序法進行排序
 2. 輸出須包含原陣列以及排序後的陣列
- 陣列排序結果應為數字由小到大
- 參考答案

```
before sorting
8 4 12 2 6 10 14 1 3 5
after sorting
1 2 3 4 5 6 8 10 12 14
```

- 作業程式：[hw]merge_sort.c



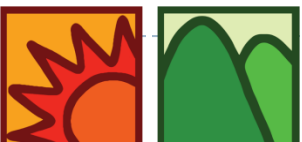


上機練習(三)

- 題目給定一個陣列，完成以下問題：
 1. 使用快速排序法進行排序
 2. 輸出須包含原陣列以及排序後的陣列
- 陣列排序結果應為數字由小到大
- 參考答案

```
before sorting
27 17 37 18 3 97 56 72 44 3 25 35 21 34 52 37 12 71 46 42
after sorting
3 3 12 17 18 21 25 27 34 35 37 37 42 44 46 52 56 71 72 97
```

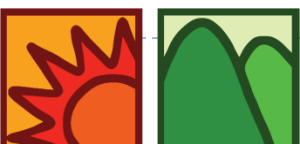
- 作業程式：[hw]quick_sort.c





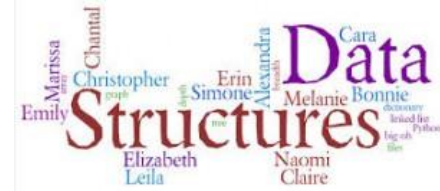
堆積 (Heap)

- 為一完全二元樹 (Complete Binary Tree)
- Max-Heap :
 - 每個節點的值會大於等於其子點的值
 - 樹根 (root) 的值會是該棵 Heap tree 中最大的
- Min-Heap :
 - 每個節點的值會小於等於其子點的值
 - 樹根 (root) 的值會是該棵 Heap tree 中最小的





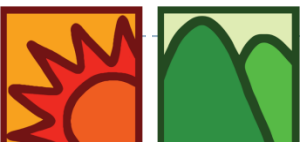
上機練習(四)



- 實作 Max heap
- 需印出每次 pop 出來的元素
- 參考答案 →
- 作業程式：[hw]max_heap.c

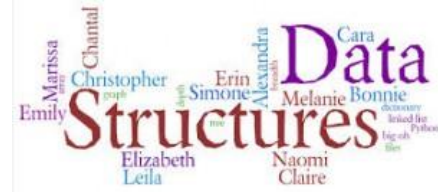
```

=====Print Heap=====
9 -> 8 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1
Pop Max : 9
=====Print Heap=====
8 -> 6 -> 7 -> 2 -> 5 -> 4 -> 3 -> 1
Pop Max : 8
=====Print Heap=====
7 -> 6 -> 4 -> 2 -> 5 -> 1 -> 3
Pop Max : 7
=====Print Heap=====
6 -> 5 -> 4 -> 2 -> 3 -> 1
Pop Max : 6
=====Print Heap=====
5 -> 3 -> 4 -> 2 -> 1
Pop Max : 5
=====Print Heap=====
4 -> 3 -> 1 -> 2
Pop Max : 4
=====Print Heap=====
3 -> 2 -> 1
Pop Max : 3
=====Print Heap=====
2 -> 1
Pop Max : 2
=====Print Heap=====
1
Pop Max : 1
=====Print Heap=====
__Heap is Empty__
  
```

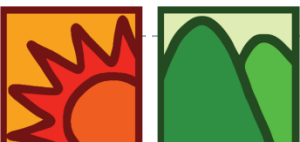




上機注意事項

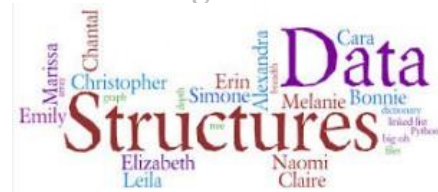


- 一律使用 C 語言撰寫!
- 上機題目只要寫好其中一題，就可以先拿給助教檢查、登記，不用等到全部寫完!
- 若課堂上未完成，可當作回家作業，再於後兩周的助教時間拿給助教檢查即可!
- 程式碼不須繳交，只須當下 **DEMO** 給助教看就好!
- 若無法解釋code邏輯，不予計分!
- 若無法於助教時間繳交作業，請提前一天以前告知!
- 遲交一天分數 *0.8，兩天分數 *0.6，之後不受理補交!





其他事項



助教時間 (地點: CM 3038)

1. 5/24 (三) 19:00 ~ 20:00
2. 5/31 (三) 19:00 ~ 20:00

助教聯絡方式：

林妤嫻：

yulei3847@gmail.com

謝博丞：

pocheng0301@gmail.com

陳亞琦：

m104020042@g-mail.nsysu.edu.tw

