



排序

排序的目的之一

- 較容易閱讀
- 利於統計及整理
- 可大幅減少數據搜尋的時間



5個整數，
由小到大排列。

問題



可以這樣比...

原始資料	第一回合				第二回合				第三回合				第四回合				
4	3	3	3	3	2	2	2	2	2	2	2	2	2	1	繼續往下比...		
3	4	2	2	2	3	3	3	3	3	1	1	1	1	2			
2	2	4	4	4	4	4	1	1	1	3	3	3	3	3			
5	5	5	5	1	1	1	4	4	4	4	4	4	4	4			
1	1	1	1	5	5	5	5	5	5	5	5	5	5	5			

方法:

比4大回合，

每一回合又比4小回合

兩兩相比，小的往上提



將方法變成程式碼

```
#include <stdio.h>
int main() {
    int array[11];
    int i,j,temp;
    for (i=0;i<5;i++)
        scanf("%d",&array[i]);

    for (i=0;i<5;i++){ //比4大回合
        for(j=0;j<5;j++){//比4小回合
            if(array[j]>array[j+1]){
                temp=array[j];
                array[j]=array[j+1];
                array[j+1]=temp;
            }
        }
    }

    for (i=0;i<5;i++)
        printf("%d ",array[i]);

    return 0;
}
```

輸入資料

前一個比後一個大，就做交換

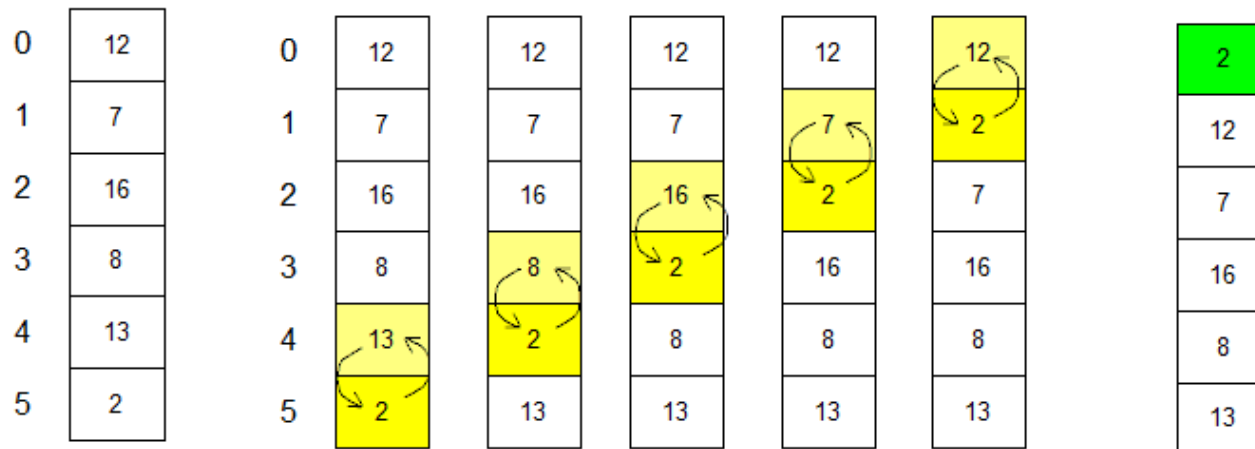
輸出資料



這是泡沫排序法

- 重複地走訪要排序的數列，一次比較兩個元素，如果他們的順序錯誤就把他們交換過來。
- 走訪數列的工作是重複地進行直到沒有再需要交換，也就是說該數列已經排序完成。
- 越小的元素會經由交換慢慢「浮」到數列的頂端。

CS50說氣泡





還有其他方法嗎？

也可以這樣比

原始資料	Step 1	Step 2	Step 3
4	1	1	1
3	4	2	2
2	3	4	3
5	2	3	4
1	5	5	5

方法:
在每一回合中找出最小值



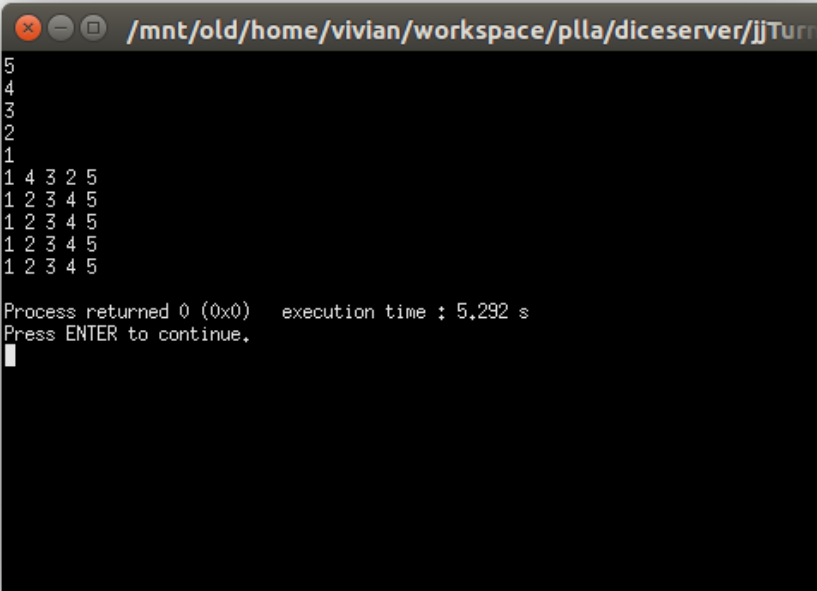
將方法變成程式碼

歡迎參考 [edX 上開授的 CS 50 課程](#) 示範影片:

```
#include <stdlib.h>
#include <stdio.h>
int main() {
    int num[10], flag;
    int i, j, temp=0, t;
    for(i=0; i<5; i++){
        scanf("%d", &num[i]);
    }

    for(i=0; i<5; i++){
        flag=i;
        for(j=i+1; j<5; j++){
            if(num[flag]>num[j]){
                flag=j; //找到最小值的位置
            }
        }
        temp=num[flag];
        num[flag]=num[i];
        num[i]=temp;
    }
    for(t=0; t<5; t++){
        printf("%d ", num[t]);
    }
    printf("\n");
}

return 0;
}
```



這是選擇排序法

- 在所有的資料中，
當由大至小排序，則將最大值放入第一位置；
若由小至大排序時，則將最大值放入位置末端。
- 例如當N筆資料需要由大至小排序時，首先以第一個位置的資料，依次向2、3、4 ...N個位置的資料作比較。
- 如果資料大於或等於其中一個位置，則兩個位置的資料不變；
若小於其中一個位置，則兩個位置的資料互換。



排序，還有很多種方法呢!

- 內部排序：
排序的資料量小，可以完全在記憶體內進行排序。
 - 氣泡排序法
 - 選擇排序法
 - 插入排序法
 - 合併排序法
 - 快速排序法
 - 堆積排序法
 - 謝耳排序法
 - 基數排序法
- 外部排序：
排序的資料量無法直接在記憶體內進行排序，而必須使用到輔助記憶體
 - 直接合併排序法
 - k路合併法
 - 多相合併法

