

氣泡排序法

泡泡排序法

1. 請使用泡泡排序法(Bubble Sort)撰寫程式。
2. 程式內有一鍵盤輸入的資料陣列。
3. 請輸出泡泡排序法(由大排到小)的比對過程。
4. 執行結果如下所顯示。

```
請輸入5個整數：3 5 2 8 7
5 3 8 7 2
5 8 7 3 2
8 7 5 3 2
8 7 5 3 2
```

```
請輸入5個整數：5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
```

氣泡排序法(Bubble Sort)

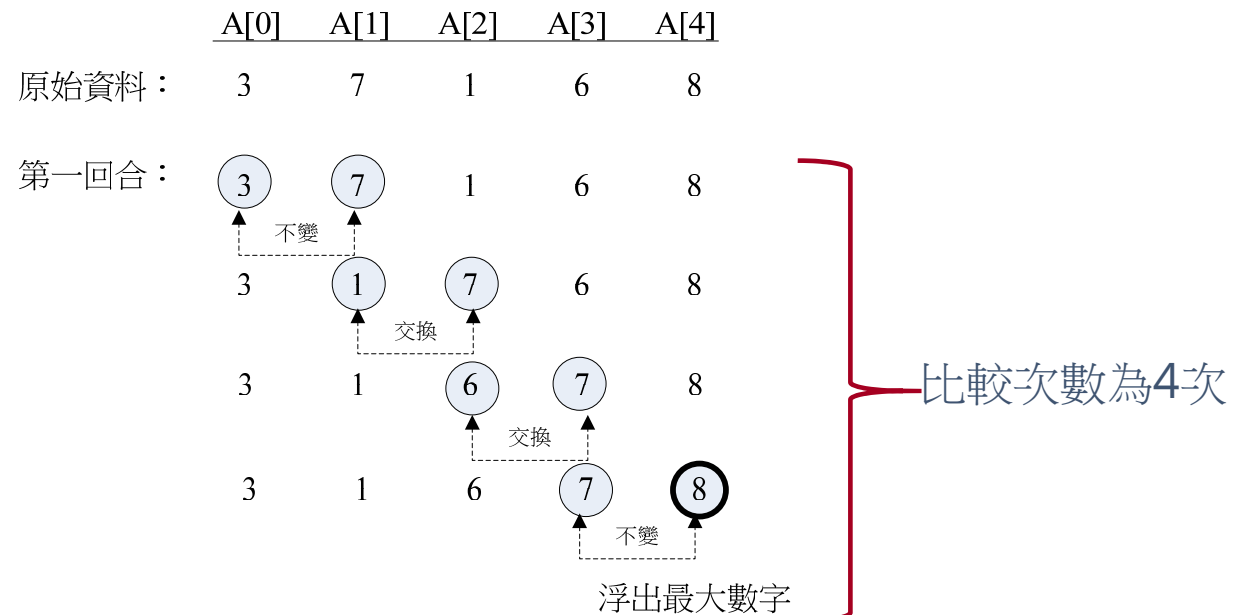
【定義】

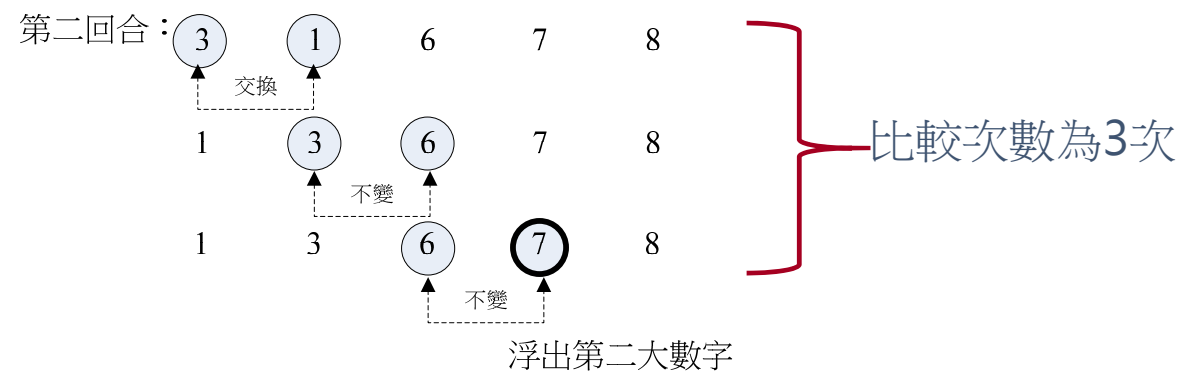
是指將兩個相鄰的資料相互做比較，若比較時發現次序不對，則將兩個資料互換，並且資料依序由上往下比，而結果則會依序由下往上浮起，猶如氣泡一般。

【原理】

由小到大
or
由大到小

逐次比較兩個相鄰的資料，按照排序的條件交換位置，直到全部資料依序排好為止。其排序過程。如下圖所示：






```

import java.util.Scanner;
public class e {
    public static Scanner scn = new Scanner(System.in);
    public static void main(String args[]) {
        System.out.print("請問要輸入幾個整數：");
        int n=scn.nextInt();//共輸入n個元素
        int a, data[] = new int[n];

        for(a=0; a<n;a++)
            data[a] = scn.nextInt();

        int temp; //用來交換元素的暫存變數

        for(int i=0;i<data.length-1;i++) //共需進行:n-1輪
        {
            for(int j=0;j<data.length-1-i;j++ ) //第i輪比對n-i次
            {
                if(data[j] > data[j+1]) //若目標是陣列由小而大，則j>j+1的時候要彼此交換
                {
                    temp = data[j];
                    data[j] = data[j+1];
                    data[j+1] = temp;
                }
            }

            for(int k=0;k<data.length;k++)
            {
                System.out.print(data[k]+" "); //印出每一輪比較結果
            }

            System.out.println("");
        }
    }
}

```

請問要輸入幾個整數：5

3 7 1 6 8

3 1 6 7 8

1 3 6 7 8

1 3 6 7 8

1 3 6 7 8

	陣列	A[0]	A[1]	A[2]	A[3]	A[4]	說明 n=5
	輸入	3	7	1	6	8	目標：1 3 6 7 8 方法：兩兩比較
共需要 比較n-1 輪	第1輪	<u>3</u> 7168 <u>3</u> 1768 31678 <u>3</u> 1 <u>6</u> 78 ← 第1輪比較結束，印出結果(1st出現)					第i輪，需要比較n-i次
	第2輪	<u>1</u> 3 <u>6</u> 7 8 <u>1</u> 3 <u>6</u> 7 8 <u>1</u> 3 <u>6</u> 7 8 ← 第2輪比較結束，印出結果(2nd出現)					
	第3輪	<u>1</u> 3 <u>6</u> 78 <u>1</u> 3 <u>6</u> 78 ← 第3輪比較結束，印出結果(3rd出現)					
	第4輪	<u>1</u> 3 678 ← 第4輪比較結束，印出結果(4th出現)					

⊕ 氣泡排序法的演算法如下：

```
01 Procedure BubSort(int A[], int n)
02   begin
03     for (i=n-1; i>=1; i--)           //排序n-1個回合
04     {
05       for (j =0; j <=i-1; j++)       //從第0個元素開始掃描
06       if (A[j] > A[j+1])             //判斷左邊元素是否大於右邊元素
07       {                               // A[j] 與 A[j+1]交換
08         Temp = A[j];
09         A[j] = A[j+1];
10         A[j+1] = Temp;
11       }
12     }
13   end
14 End Procedure
```

【實例】

假設原始資料為：3,7,1,6,4，在進行排序時，每一回合必定會有一個元素排到定位，稱為一個回合(Pass)。

	A[0]	A[1]	A[2]	A[3]	A[4]	比較次數	比較範圍
原始資料	3	7	1	6	4		
Pass 1	3	1	6	4	7	4	(A[0]與A[1]、A[1]與A[2]、A[2]與A[3]、A[3]與A[4])
Pass 2	1	3	4	6	7	3	(A[0]與A[1]、A[1]與A[2]、A[2]與A[3])
Pass 3	1	3	4	6	7	2	(A[0]與A[1]、A[1]與A[2])
Pass 4	1	3	4	6	7	1	(A[0]與A[1])
總比較次數						10	

【分析】

1. 比較之回合次數=資料個數-1

(例如：資料個數 $n=5$ ，則回合次數為4)

2. 在每一回合之後，至少會有一個資料可以排列到正確位置，再進行下一個回合的排列時，便可以減少此資料的比較。

(例如：資料個數 $n=5$ ，則Pass 1時，比較次數為4，Pass 2時，比較次數為3，以此類推，如上表所示)

3. 需要一個額外空間。

例如：在上面的演算法中的行號08，需要一個Temp變數空間。

4. 為一種穩定排序 (Stable Sorting) 。

因為氣泡排序法交換條件為「左大右小」時才必須交換。如下所示：

(1)排序前：

3	7	1	6	4
---	---	---	---	---

(2)排序後：

1	3	4	6	7
---	---	---	---	---