

冒泡排序

冒泡排序 (Bubble Sort) 是一种计算机科学领域的较简单的排序算法，它重复地走访要排序的元素列，依次比较两个相邻的元素，如果他们的顺序 (如从大到小) 错误就把他们交换过来。走访元素的工作是重复地进行直到没有相邻元素需要交换，也就是说该元素已经排序完成。这个算法的名字由来是因为越大的元素会经由交换慢慢“浮”到数列的顶端 (升序或降序排列)，就如同碳酸饮料中二氧化碳的气泡最终会上浮到顶端一样，故名“冒泡排序”。

冒泡排序算法的原理如下：

1. 比较相邻的元素。如果第一个比第二个大，就交换他们两个。
2. 对每一对相邻元素做同样的工作，从开始第一对到结尾的最后一对。在这一点，最后的元素应该会是最大的数。
3. 针对所有的元素重复以上的步骤，除了最后一个。
4. 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较

冒泡排序算法复杂度：
假设列表有元素n个，若列表的初始状态本身就是正序的的，那么一趟扫描即可完成排序。所需的关键字比较次数c和记录移动次数m均达到最小值：即 $C = n - 1$ 和 $M = 0$ ，所以在这种状态下，算法的复杂度是线性的 $O(n)$

假设列表有元素n个，并且是逆序的，一个数列 5 4 3 2 1 进行冒泡升序排列，

第一次大循环从第一个数(5)开始，比较过程：先比较5和4，4比5小，交换位置变成4 5 3 2 1；比较5和3，3比5小，交换位置变成4 3 5 2 1；最后比较5和1，1比5小，交换位置变成4 3 2 1 5。这时候共进行了4次比较交换运算，最后1个数变成了此次排序最大数。

第二次大循环从第一个数(4)开始，比较过程：先比较4和3，3比4小，交换位置变成 3, 4, 2, 1, 5；比较4和2，2比4小，交换位置变成 3, 2, 4, 1, 5，最后比较 4和1，1比4小，交换位置变成 3, 2, 1, 4, 5。这时进行了3次比较交换运算，倒数第二数变成了此次排序最大的数。

第三次大循环从第一个数(3)开始，比较过程：先比较3和2，2比3小，交换位置变成2, 3, 1, 4, 5；比较3和1，1比3小，交换位置变成 2, 1, 3, 4, 5，这时进行了2次比较运算，倒数第三个数变成了此次排序最大的数

第四次大循环从第一个数(2)开始，比较过程：先比较2和1，1比2小，交换位置变成1, 2, 3, 4, 5 此时进行了1次比较运算。排序成功。

总共比较次数是 $4 + 3 + 2 + 1$ ，可见这是一个等差数列，等差数量的前N项和公式 $S_n = n(a_1 + a_n) / 2 = 4(4 + 1) / 2 = 10$ 次

可见如果有x个元素的列表，总共要经过x-1次循环，第一次循环的比较运算次数是x - 1，通过冒泡排序最多需要通过 $(x - 1)(x - 1 + 1) / 2 = x(x - 1) / 2$ ，按照计算复杂度的原则，去掉常数，去掉最高项系数，最终等于x^2，所以冒泡排序的时间复杂度为 x^2

冒泡排序的时间复杂度是指数级。

In [3]: #示例代码1:

```
l = [5, 1, 6, 2, 10, 0]

def fl(l):
    listLen = len(l)
    for j in range(listLen - 1): #总共循环多少次
        for k in range(listLen - 1): #元素交换的次数
            a = l[k]
            b = l[k+1]
            if l[k] > l[k+1]:
                l[k+1] = a
                l[k] = b
        listLen -= 1
    return l
print(fl(l))
```

[0, 1, 2, 5, 6, 10]

In []: