

# 冒泡排序

冒泡排序过程：

第1次交换

第2次交换

第3次交换

第4次交换

第1次排序：

3	3	3	3	1
4	4	4	1	3
5	5	1	4	4
2	1	5	5	5
1	2	2	2	2

第2次排序：

1	1
3	2
4	3
5	4
2	5


第3次排序：

1	1
2	2
3	3
4	4
5	5

5比4大  
不交换

第4次排序：

1	1
2	2
3	3
4	4
5	5



由上可知：

1. 需要4次排序

2. 每次排序需要4, 3, 2, 1次比较/交换

所以采用双重循环解决问题

1. 外层循环*i*为第几次排序

2. 内层循环*j*为第几次比较/交换

Python代码：

```
#!/usr/bin/python
```

由代码可知：

```
len = 5
```

```
a = [3,4,5,2,1]
```

```
for i in range(len-1):
```

```
    for j in range(len-1-i):
```

```
        if a[len-1-j] < a[len-1-j-1]:
```

```
            a[len-1-j], a[len-1-j-1] = a[len-1-j-1], a[len-1-j]
```

```
print a
```

1. 空间复杂度为 $O(1)$

2. 时间复杂度为 $O(n^2)$ ：

统计循环次数为：

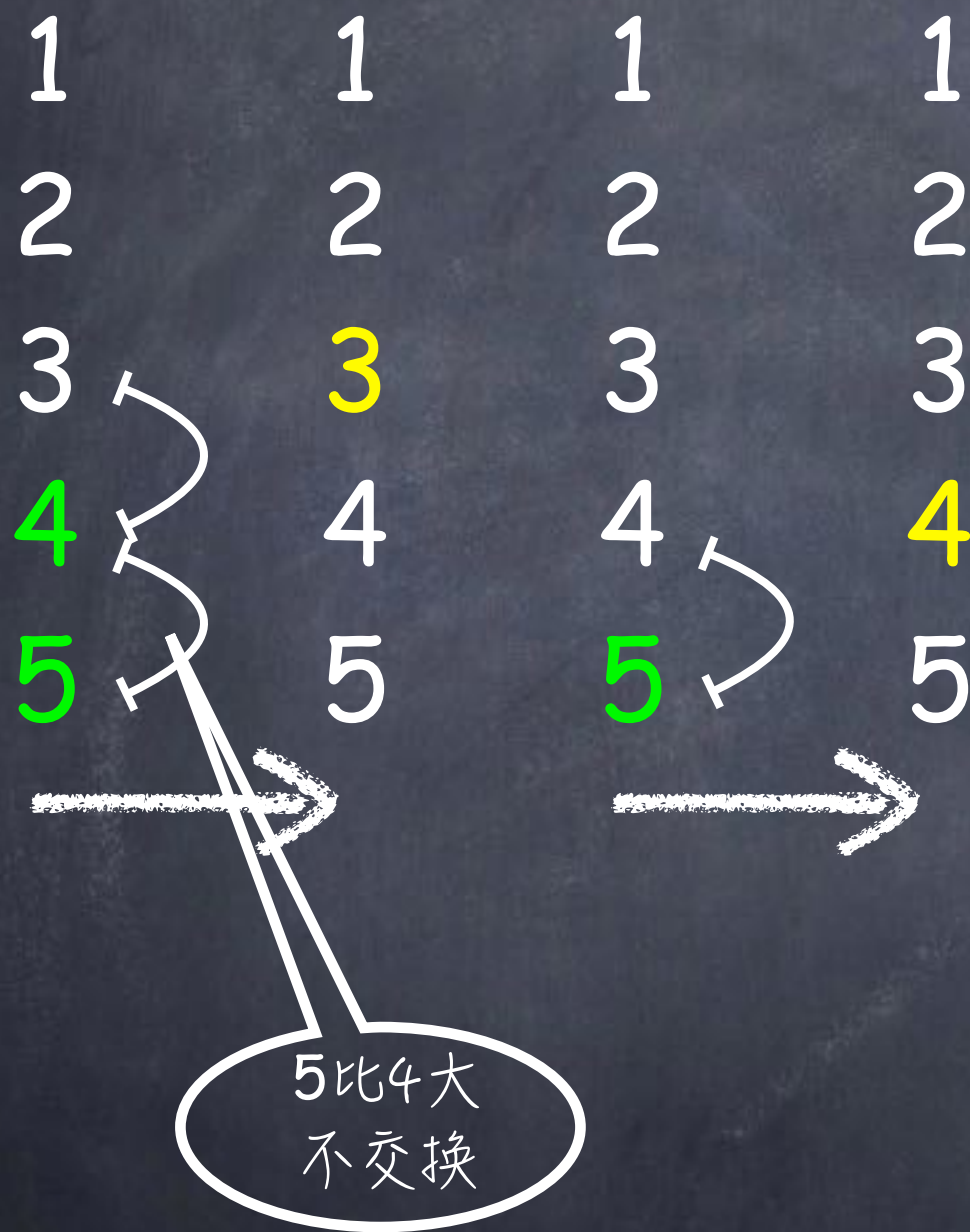
$4+3+2+1 \Rightarrow n*(n-1)/2$

$O(n*(n-1)/2) \sim O(n^2)$



# 冒泡排序简单优化

仔细查看第3、4次排序



如果第3次排序，没有发生交换  
那么，后续数列为有序，则可以  
直接结束

Python代码：

```
#!/usr/bin/python
len = 5
a = [3,4,5,2,1]
flag = 1 #默认为有序
for i in range(len-1):
    flag = 1
    for j in range(len-1-i):
        if a[len-1-j] < a[len-1-j-1]:
            a[len-1-j], a[len-1-j-1] = a[len-1-j-1], a[len-1-j]
            flag = 0 #如果发生交换设置为无序
    if flag == 1 :
        break;
print a
```