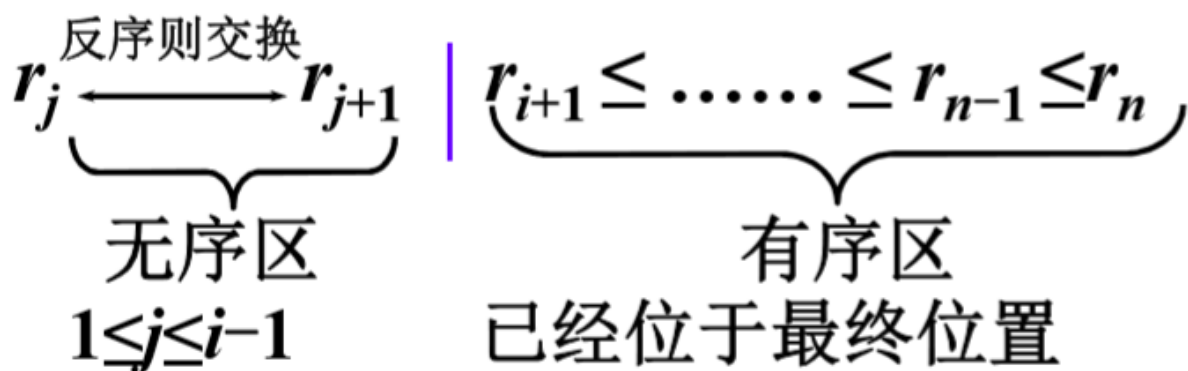


# 冒泡排序

## 基本思想

两两比较相邻记录的关键码，如果反序则交换，直到没有反序的记录为止



需要解决的问题：

- 在一趟起泡排序中，若有多个记录位于最终位置，应如何记载
- 如何确定起泡排序的范围，使得已经位于最终位置的记录不参与下一趟排序？
- 如何判别起泡排序的结束？

第一个问题：

设变量exchange记载记录交换的位置，则一趟排序后，exchange记载的一定是这一趟排序中记录的最后一次交换的位置，且从此位置以后的所有记录均已经有序。

第二个问题：

设bound位置的记录是无序区的最后一个记录，则每趟起泡排序的范围是 $r[1] \sim r[\text{bound}]$ 。在一趟排序后，从exchange位置之后的记录一定是有序的，所以 $\text{bound} = \text{exchange}$ 。

第三个问题：

在每一趟起泡排序之前，令exchange的初值为0，在以后的排序过程中，只要有记录交换，exchange的值就会大于0。这样，在一趟比较完毕，就可以通过exchange的值是否为0来判别是否有记录交换，从而判别整个起泡排序的结束。

代码：

```
#include <iostream>

using namespace std;

class BubbleSort{
public:
    void sort(int nums[], int n){
        int exange = 1, bound = n;
        while(exange){
            exange = 0;
            for(int i=1;i<bound;i++){
                if(nums[i] < nums[i-1]){
                    swap(nums[i], nums[i-1]);
                    exange = i;
                }
            }
            bound = exange;
        }
    }
};

int main(){
    BubbleSort sort;
    int nums[]={3,2,0,-1,5};
    sort.sort(nums, 5);
    for(int i=0;i<5;i++)
        cout<<nums[i]<<" ";
    cout<<endl;
}
```