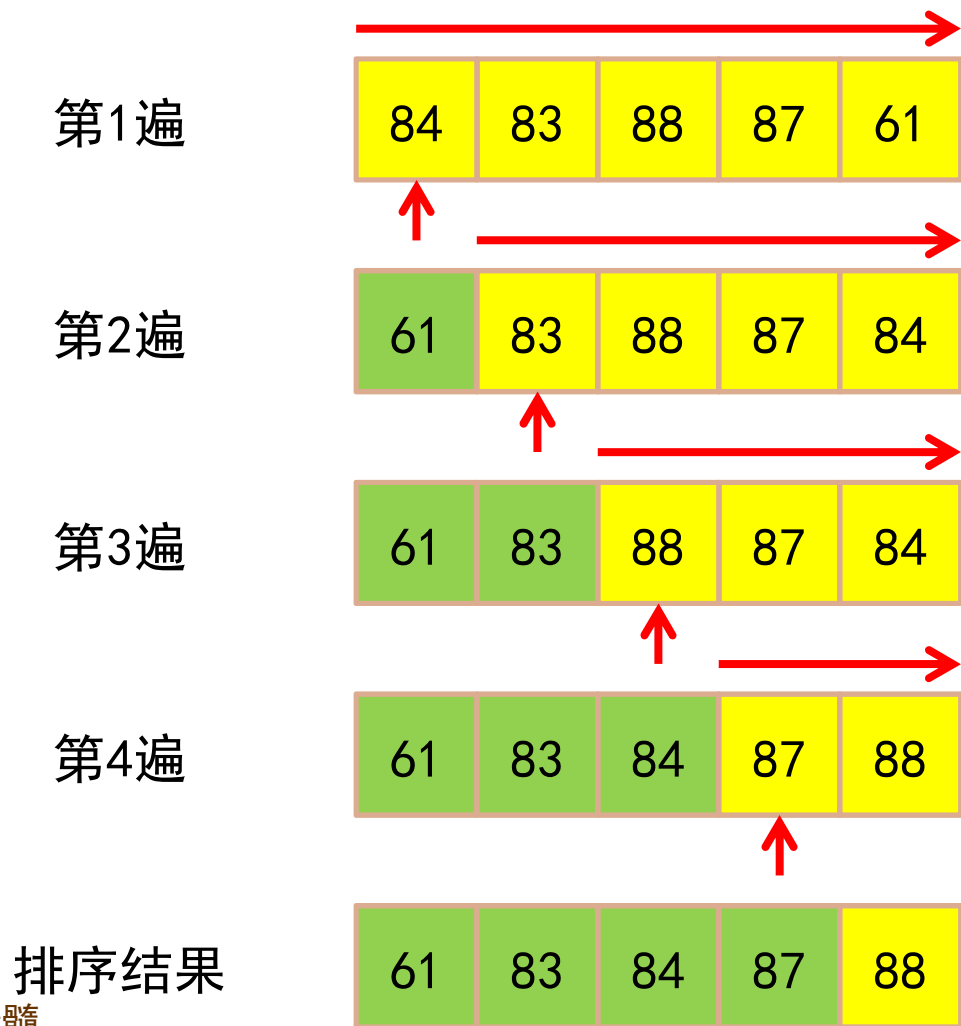


# 选择法排序



在每一遍比较中，在剩余的待比较的数中选择一个最小的数与这个剩余序列的第1个数交换位置

# 选择法排序

```
for (i=0; i<n-1; i++)  
{
```

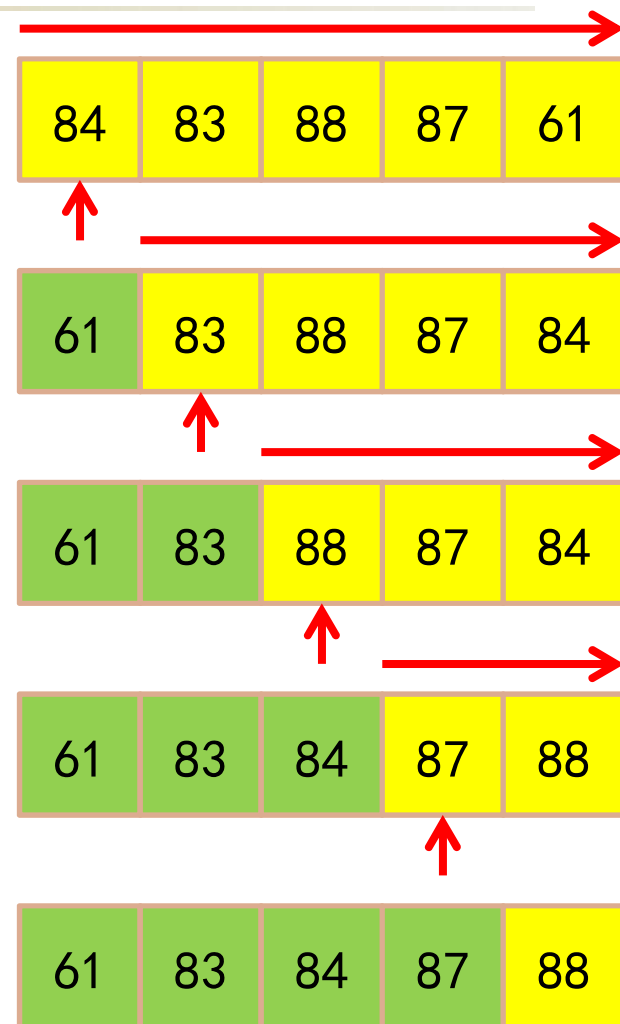
寻找最低分所在下标k的过程

```
    k = i;  
    for (j=i+1; j<n; j++)  
    {  
        if (score[j] < score[k])  
            记录此轮比较中最低分  
            所在元素的下标 k = j;  
    }
```

若k中记录的最低分位置不在下标i处，则

"交换成绩score[k]和score[i]"

```
}
```



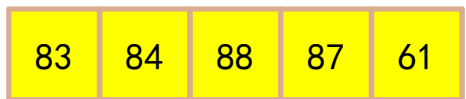
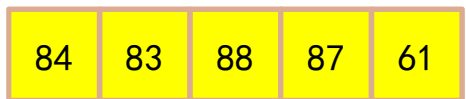
# 选择法排序

```
void SelectionSort(int score[], int n) /*选择法*/
{
    int i, j, k, temp;
    for (i=0; i<n-1; i++)
    {
        k = i;
        for (j=i+1; j<n; j++)
        {
            if (score[j] < score[k])
            {
                k = j; /*记录最小数下标位置*/
            }
        }
        if (k != i) /*若最小数不在下标位置i*/
        {
            temp = score[k];
            score[k] = score[i];
            score[i] = temp;
        }
    }
}
```

成绩升序排序

# 冒泡法排序

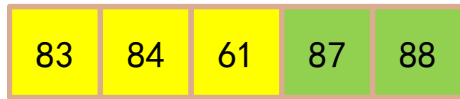
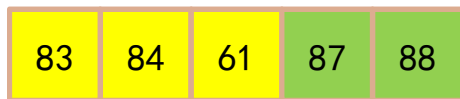
第1遍



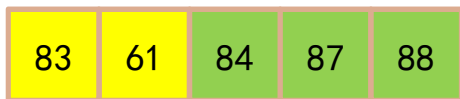
第2遍



第3遍



第4遍



之所以称其为冒泡排序，是因为算法中值相对较小的数据会像水中的气泡一样逐渐上升到数组的最顶端

比较相邻的两个数据  
若顺序不对，则将其位置交换

# 冒泡法排序

```
void BubbleSort(int score[], int n)
{
    int i, j, temp;
    for (i=0; i<n-1; i++)
    {
        for (j=1; j<n-i; j++)
        {
            if (score[j] < score[j-1])
            {
                temp = score[j];
                score[j] = score[j-1];
                score[j-1] = temp;
            }
        }
    }
}
```

交换相邻元素

有改进的方法吗？

