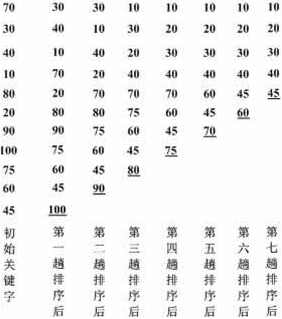
# 交换排序

介绍两种经典的交换排序——冒泡排序和快速排序。

## 一、冒泡排序

冒泡排序是一种交换排序，它的主要过程是：

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。比较一趟之后，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。



见 冒泡排序演示1

冒泡排序的代码实现：

#include <stdio.h>#include <stdlib.h>

int n;

/\*

\* 冒泡排序

\*/void BubbleSort(int \*array){

int i, j, temp;

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

{

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

{

if (array[j] > array[j + 1])

{

temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

}

int main(){

int i;

int \*array;

printf("请输入数组的大小：");

scanf("%d", &n);

array = (int\*) malloc(sizeof(int) \* n);

printf("请输入数据（用空格分隔）：");

for (i = 0; i < n; i++)

{

scanf("%d", &array[i]);

}

BubbleSort(array);

printf("排序后为：");

for (i = 0; i < n; i++)

{

printf("%d ", array[i]);

}

printf("\n");

}

## 二、快速排序

快速排序是对冒泡排序的改进，它的基本思想是通过一趟排序将数据分成两部分，一部分中的数据都比另一部分中的数据小，再对这两部分中的数据再排序，直到整个序列有序，如下图所示。

见 快速排序演示

快速排序的代码实现：

#include <stdio.h>#include <stdlib.h>

int n;

/\*

\* 分割使枢轴记录的左边元素比右边元素小

\*/int Partition(int \*array, int low, int high){

int pivotkey = array[low];

array[0] = array[low];

while (low < high)

{

while (low < high && array[high] >= pivotkey)

{

high--;

}

array[low] = array[high];

while (low < high && array[low] <= pivotkey)

{

low++;

}

array[high] = array[low];

}

array[low] = array[0];

return low;

}

/\*

\* 快速排序递归实现

\*/void QuickSort(int \*array, int low, int high){

if (low < high)

{

int pivotloc = Partition(array, low, high);

QuickSort(array, low, pivotloc - 1);

QuickSort(array, pivotloc + 1, high);

}

}

int main(){

int i;

int \*array;

printf("请输入数组的大小：");

scanf("%d", &n);

array = (int\*) malloc(sizeof(int) \* (n + 1));

printf("请输入数据（用空格分隔）：");

for (i = 1; i <= n; i++)

{

scanf("%d", &array[i]);

}

QuickSort(array, 1, n);

printf("排序后为：");

for (i = 1; i <= n; i++)

{

printf("%d ", array[i]);

}

printf("\n");

}

## 三、小结

这一章讲了交换排序的两个经典算法，冒泡排序和快速排序。冒泡排序就像水中的气泡一样，小的数据往上浮，它是稳定的排序，它的时间复杂度是O(n^2)。快速排序是对冒泡排序的改进，它的主要思想是通过一趟排序将数据分成两部分，一部分中的数据都比另一部分中的数据小，再对这两部分中的数据再排序，直到整个序列有序，它是不稳定的排序，它的时间复杂度是O(nlogn)。

## 作业

我们这里的快速排序是用递归实现的，那么怎么用非递归实现呢？