**冒泡排序**

js实现方式一：

for循环实现

function mp1(arr) {

for(i = 0; i < arr.length; i++ ) {

for(j = 0; j < arr.length - i -1; j++) {

if(arr[j] > arr[j+1]) {

var temp;

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

sort函数实现

function mp2(arr) {

arr.sort( function () {

return a-b;

})

}

**快速排序**

基本思想：

通过一趟排序将排记录分隔成独立的两个部分，其中一部分记录的数比另一部分记录的数 小，则可分别对这两个部分继续排序，已达到序列有序。

算法实现：

从数组中挑选一个元素，作为”基准“；

重新对数组排序，所有比”基准“小的放在”基准“左边，大的则放在右边。在这个分区退出后，该基准数就位于数列的中间位置。称为“分区”操作。

递归地把小于基准元素的子数列和大于基准值的子数列进行排序

function ks(arr , left, right) {

var i = left;

var j = right;

var temp = a[left];

var t;

if(left > right) {

return;

}

while(i != j) {

//先找右边的

while(arr[j] >= temp && i < j) {

j --

}

while(arr[i] <= temp && i < j) {

i --

}

if(i < j) {

//交换位置

t = arr[i];

arr[i] = arr[j];

arr[j] = t;

}

}

//基准数归位

arr[left] = arr[i]

arr[i] = temp

//递归

ks(arr, left, i -1); //处理左边

ks(arr, i + 1, right); //处理右边

}