1.以一个数为基准(中间的数)，比基准小的放到左边，比基准大的放到右边

2.再按此方法对这两部分数据分别进行快速排序（递归进行）

3.不能再分后退出递归，并重新将数组合并

var quickSort = function(myArray) {

// 当被分的数组只剩一个时，退出递归

if (myArray.length <= 1) {

return myArray;

}

// 中间基准值的index

var pivotIndex = Math.floor(myArray.length / 2);

// 基准值

var pivot = myArray.splice(pivotIndex, 1)[0];

var left = [];

var right = [];

// 小的放左边，大的放右边

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

if (myArray[i] < pivot) {

left.push(myArray[i]);

} else {

right.push(myArray[i]);

}

}

// 递归

// 把数组合并在一起

return quickSort(left).concat([pivot], quickSort(right));

};

重点：创建了两个新的数组left、right，对left和right不断切片直到数组长度为1，这时单个的数组就是有序的，在逐级合并有序的数组成为一个完整的有序数组。

方式2：

1. 取基准数，把基准数扔到数组中轴的位置
2. 遍历数组，比较左右两边的数与基准数的大小，小的在左，大的在右
3. 对左右无序部分，重复第二步，直到数组为长度为1的数组

示例如下:

