算法中的递归函数在数组长度小时，内存占用不大，但是当数组长度超过一定限制后，频繁的调用递归方法会导致内存无限占用而不能及时释放，影响性能。

可以考虑匿名函数的方式实现优化，因为匿名函数执行完就销毁了，会自动释放内存。

var quickSort = (function(){

return 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));

}

})();

quickSort([1,5,3,2,4])

方法解析：

1. quickSort是一个自执行的匿名函数，函数主体是

return function(myArray) {……}这部分代码，即自执行函数执行的函数体部分