**JavaScript插入排序**

1. 插入排序概述

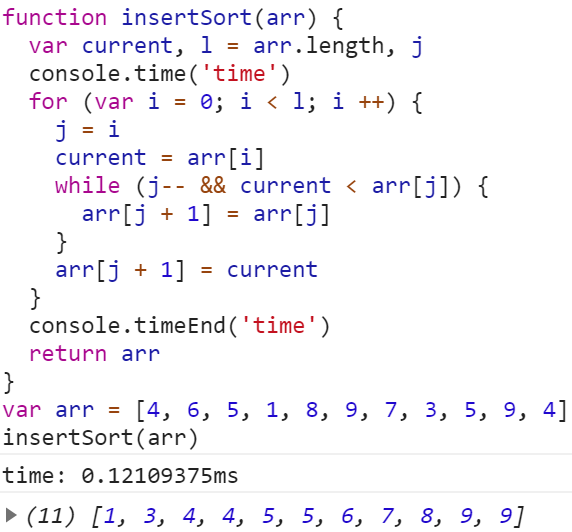
插入排序是将数组分为待排序和已排序两个区间。依次从待排序区间中取出一项，用该项跟已排序区间项逐个对比，通过位移来实现插入到对应位置的排序方式。插入排序平均时间复杂度是：O(n^2)

步骤是：

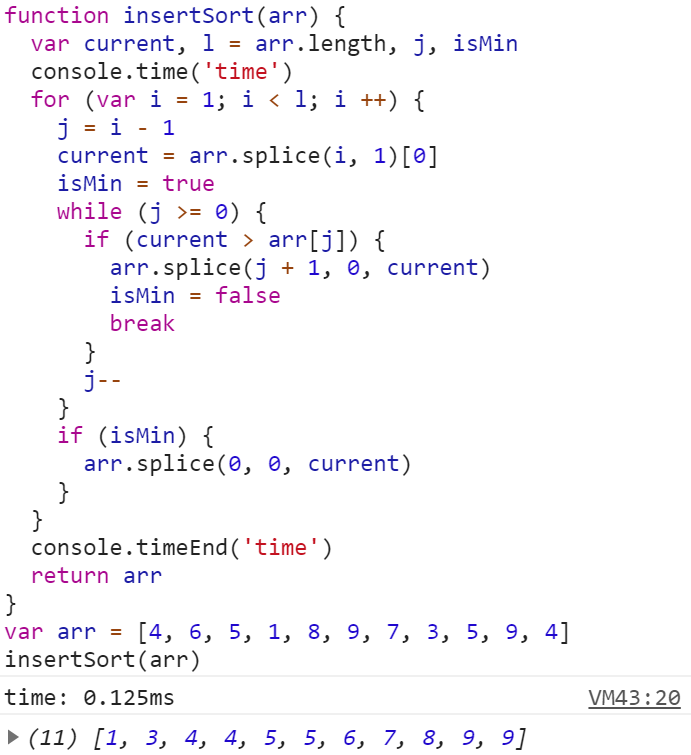
1. 先建立两个循环，外循环用于遍历待排序区间，内循环用来遍历已排序区间。
2. 从待排序中按顺序取出一项暂存起来，将该项自右往左与已排序项逐个对比，当遇到比自己大的项(表示升序)时，将该位置右移1位。
3. 再将待排序区间里右移后空出来的位置赋值为暂存项。此时已排序区间增加了一项，待排序区间减少了一项，继续第2步直到待排序遍历完成。

插入排序有多种实现方式，这里介绍常见的3种：

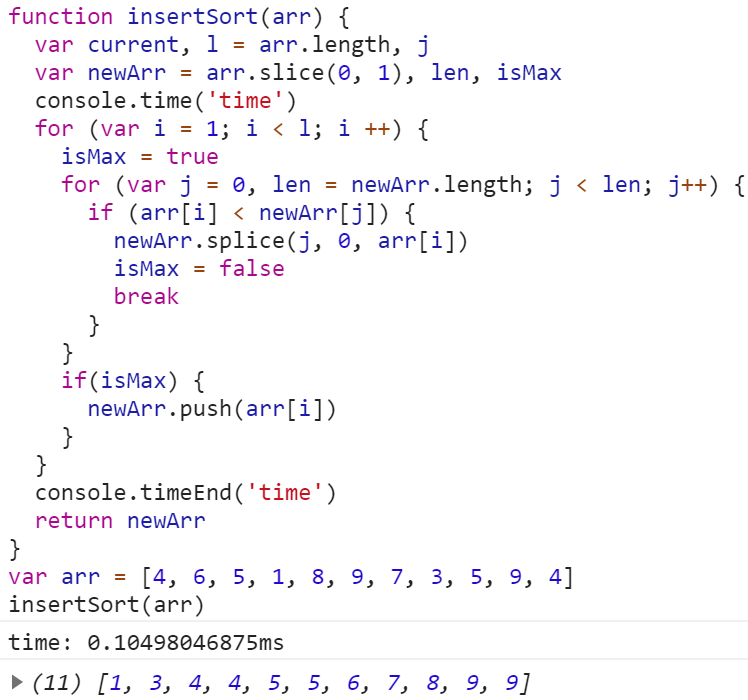
1. 插入排序实现
   1. 通用实现，自左往右遍历待排序数组，再从当前的左侧位置开始自右往左循环已排序数组，再逐个比较和移动被比较项，最后将当前项填入到空缺位置上。

­­­­­­

* 1. 利用数组splice方法，类似打扑克牌，先拿出要排序的牌，然后找准位置插入。这种方式利用了原生API，减少了数组反复移动位置的操作。性能上之前差不多。



* 1. 新建数组法与splice结合法，这种方式会多建立一个数组，也就会多占用一个空间，但理解起来最容易。



源码：

// 标准版

function insertSort(arr) {

var current, l = arr.length, j

console.time('time')

for (var i = 0; i < l; i ++) {

j = i

current = arr[i]

while (j-- >= 0 && current < arr[j]) {

arr[j + 1] = arr[j]

}

arr[j + 1] = current

}

console.timeEnd('time')

return arr

}

var arr = [4, 6, 2, 1, 8, 9, 7, 3, 5, 9, 4]

insertSort(arr)

// 标准通用版

function insertSort(arr) {

var current, l = arr.length, j

console.time('time')

for (var i = 0; i < l; i ++) {

j = i - 1

current = arr[i]

while (j >= 0 && current < arr[j]) {

arr[j + 1] = arr[j]

j--

}

arr[j + 1] = current

}

console.timeEnd('time')

return arr

}

var arr = [4, 6, 5, 1, 8, 9, 7, 3, 5, 9, 4]

insertSort(arr)

// splice版本

function insertSort(arr) {

var current, l = arr.length, j, isMin

console.time('time')

for (var i = 1; i < l; i ++) {

j = i - 1

current = arr.splice(i, 1)[0]

isMin = true

while (j >= 0) {

if (current > arr[j]) {

arr.splice(j + 1, 0, current)

isMin = false

break

}

j--

}

if (isMin) {

arr.splice(0, 0, current)

}

}

console.timeEnd('time')

return arr

}

var arr = [4, 6, 5, 1, 8, 9, 7, 3, 5, 9, 4]

insertSort(arr)

// 新建数组版

function insertSort(arr) {

var current, l = arr.length, j

var newArr = arr.slice(0, 1), len, isMax

console.time('time')

for (var i = 1; i < l; i ++) {

isMax = true

for (var j = 0, len = newArr.length; j < len; j++) {

if (arr[i] < newArr[j]) {

newArr.splice(j, 0, arr[i])

isMax = false

break

}

}

if(isMax) {

newArr.push(arr[i])

}

}

console.timeEnd('time')

return newArr

}

var arr = [4, 6, 5, 1, 8, 9, 7, 3, 5, 9, 4]

insertSort(arr)

// 新建数组版倒序遍历

function insertSort(arr) {

var current, l = arr.length, j

var newArr = arr.slice(0, 1), len, isMin

console.time('time')

for (var i = 1; i < l; i ++) {

isMin = true

len = newArr.length

while(len--) {

if (arr[i] > newArr[len]) {

newArr.splice(len + 1, 0, arr[i])

isMin = false

break

}

}

if(isMin) {

newArr.unshift(arr[i])

}

}

console.timeEnd('time')

return newArr

}

var arr = [4, 6, 5, 1, 8, 9, 7, 3, 5, 9, 4]

insertSort(arr)