插入排序

插入排序（Insertion-Sort）的算法描述是一种简单直观的排序算法。它的工作原理是通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。插入排序在实现上，通常采用in-place排序（即只需用到O(1)的额外空间的排序），因而在从后向前扫描过程中，需要反复把已排序元素逐步向后挪位，为最新元素提供插入空间。

算法实现描述

一般来说，插入排序都采用in-place在数组上实现。具体算法描述如下：

- <1>.从第一个元素开始，该元素可以认为已经被排序；
- <2>.取出下一个元素，在已经排序的元素序列中从后向前扫描；
- <3>.如果该元素（已排序）大于新元素，将该元素移到下一位置；
- <4>.重复步骤3，直到找到已排序的元素小于或者等于新元素的位置；
- <5>.将新元素插入到该位置后；
- <6>.重复步骤2~5。

时间复杂度： 最好的情况 O(n)、最坏的情况 O(n * n)、平均情况 O(n * n)

空间复杂度: O(1)

是否稳定： 稳定

方法一实现

```
var a = [3,4,5,6,1,9,2]

function insertSort(arr) {
    if(Object.prototype.toString.call(arr).slice(8, -1) === 'Array') {
        console.time('插入排序耗时：');
        for(var i = 1; i < arr.length; i++) {
            var key = arr[i]
            var j = i - 1
            while(j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j]
                j--
            }
            arr[j + 1] = key
        }
        console.timeEnd('插入排序耗时：');
        return arr
    } else {
        new Error('array is not an Array!')
    }
}

console.log(insertSort(a))
```

改进实现方式：利用二分查找