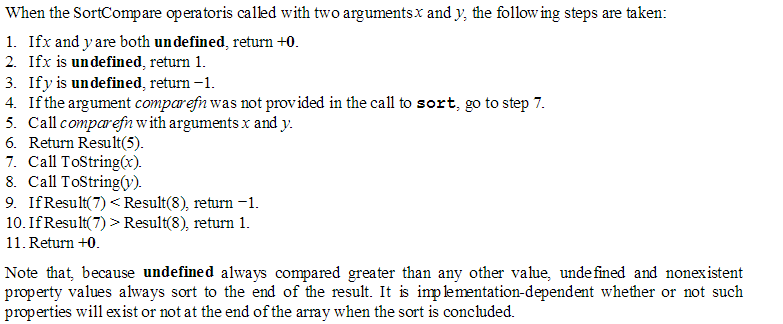
[2,20,22,21,3].sort() => [2, 20, 21, 22, 3]

排序方法中 toString 造成的



Sort 传入比较函数来获取正确的结果

[2,20,22,21,3].sort( function(a,b){ return a-b} ) => [2, 3, 20, 21, 22]

快速排序通常被认为是高效，快速等特点是使用V8引擎的实现Array.prototype.sort()上有超过23个项目的数组。 对于少于23个项目，V8采用插入排序法[2]。

归并排序是快速排序的竞争对手，因为它也是高效，快捷，但有被稳定的好处。 这就是为什么Mozilla和Safari中使用它自己的执行Array.prototype.sort()

sort默认排序顺序是根据字符串Unicode码点。如果没有指明 compareFunction ，那么元素会按照转换为的字符串的诸个字符的Unicode位点进行排序。例如 "Banana" 会被排列到 "cherry" 之前。当数字按由小到大排序时，9 出现在 80 之前，但因为（没有指明 compareFunction），比较的数字会先被转换为字符串，所以在Unicode顺序上 "80" 要比 "9" 要靠前。

如果指明了 compareFunction ，那么数组会按照调用该函数的返回值排序。即 a 和 b 是两个将要被比较的元素：

如果 compareFunction(a, b) 小于 0 ，那么 a 会被排列到 b 之前；

如果 compareFunction(a, b) 等于 0 ， a 和 b 的相对位置不变。备注： ECMAScript 标准并不保证这一行为，而且也不是所有浏览器都会遵守（例如 Mozilla 在 2003 年之前的版本）；

如果 compareFunction(a, b) 大于 0 ， b 会被排列到 a 之前。

compareFunction(a, b) 必须总是对相同的输入返回相同的比较结果，否则排序的结果将是不确定的。