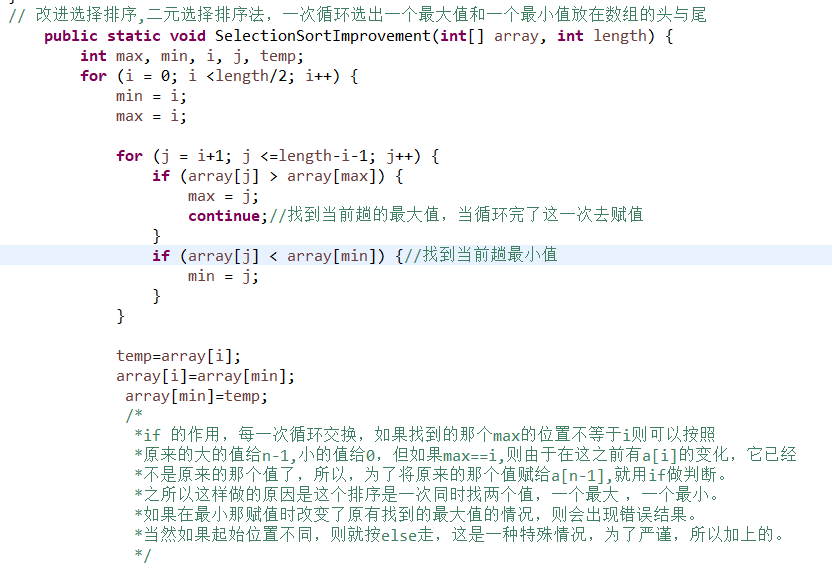
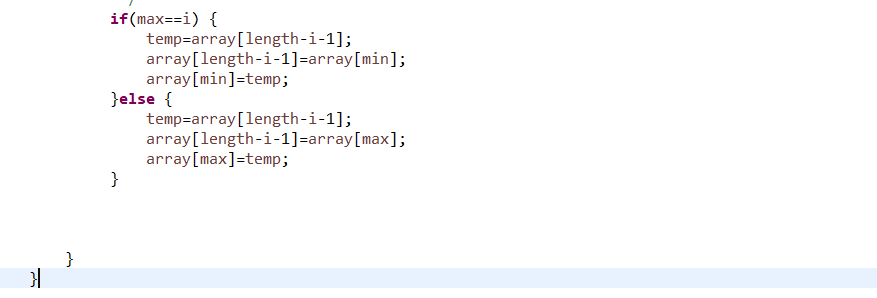
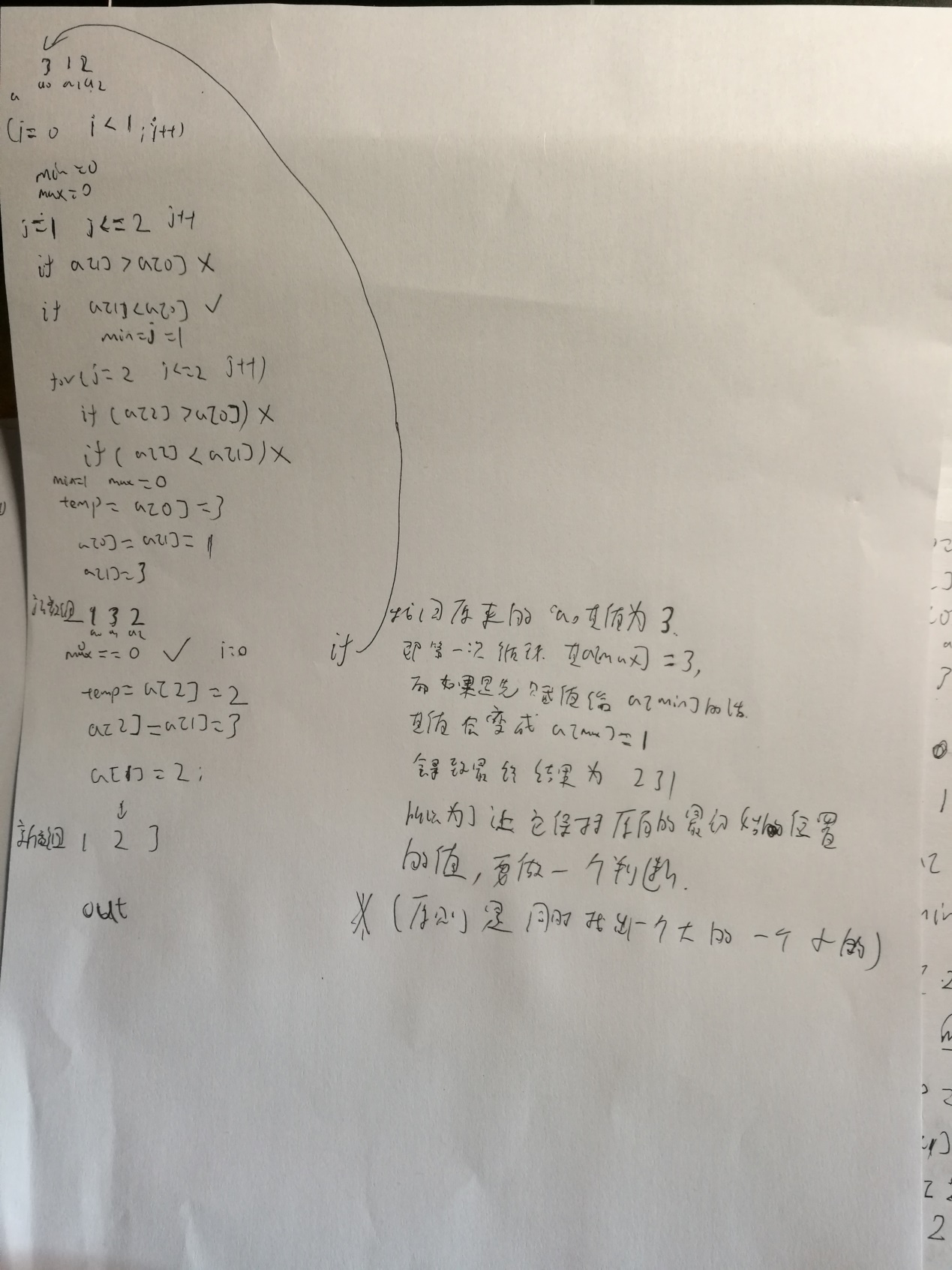
BinarySelectionSort(二元选择排序)

1. 概念：二元选择排序是在简单选择排序的基础上做的，简单选择排序每一次只选取一个最大值或最小值放到数组开始或结尾。而二元选择排序是在一次循环中同时选取一个最大值和一个最小值（当前这一趟），把它们分别放到数组开头和结尾。
2. 优点：减少排序所需的循环次数，提高效率。改进后对n个数据进行排序，最多只需进行[n/2]趟循环即可。
3. 如何实现：在n/2的循环内，设置最大位置与最小位置max、min。这两个值随循环内的i而变化。在其内嵌套的另一个循环负责用类似于打擂台的方法找到当前趟的最大值和最小值，如果先找到了最大值则用continue跳出嵌套循环进行当前趟最大值与最小值的位置交换。（此时有个注意我们一会详解），之后i++在进行下一次循环。直到循环结束，则排序结束。
4. 代码：





1. 注意：if 的作用，每一次循环交换，如果找到的那个max的位置不等于i则可以按照原来的大的值给n-1,小的值给0，但如果max==i,则由于在这之前有a[i]的变化，它已经不是原来的那个值了，所以，为了将原来的那个值赋给a[n-1],就用if做判断。之所以这样做的原因是这个排序是一次同时找两个值，一个最大 ，一个最小。如果在最小那赋值时改变了原有找到的最大值的情况，则会出现错误结果。当然如果起始位置不同，则就按else走，这是一种特殊情况，为了严谨，所以加上的。(给个例子看一下)



1. continue会跳出当前循环一次，然后再次执行该循环直到达到循环的限制条件。