# 520 第七周学习总结

先解决超哥课上布置的一个作业：用自己熟悉的语言手写各种排序算法。

我这里完成三个：选择排序、插入排序、冒泡排序。

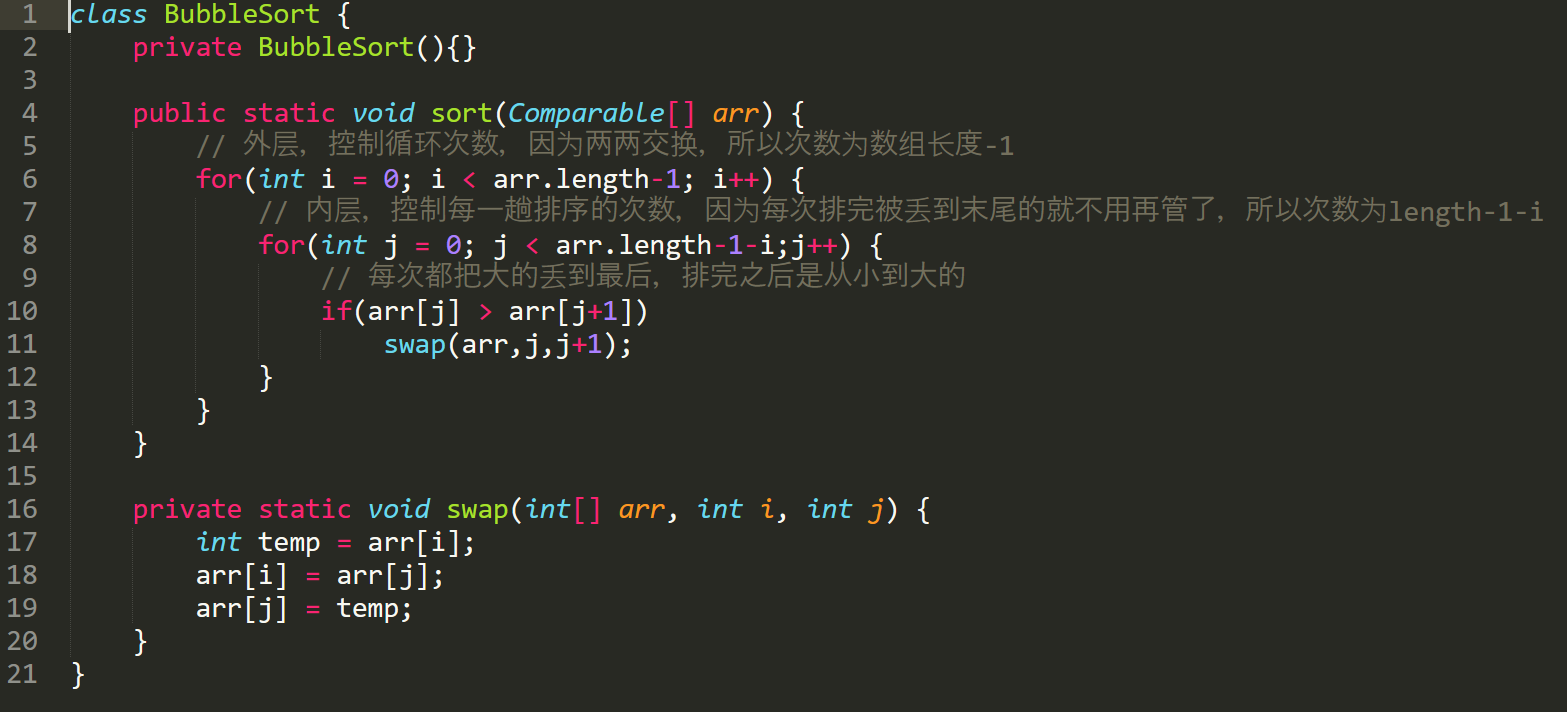
选择排序：



插入排序：

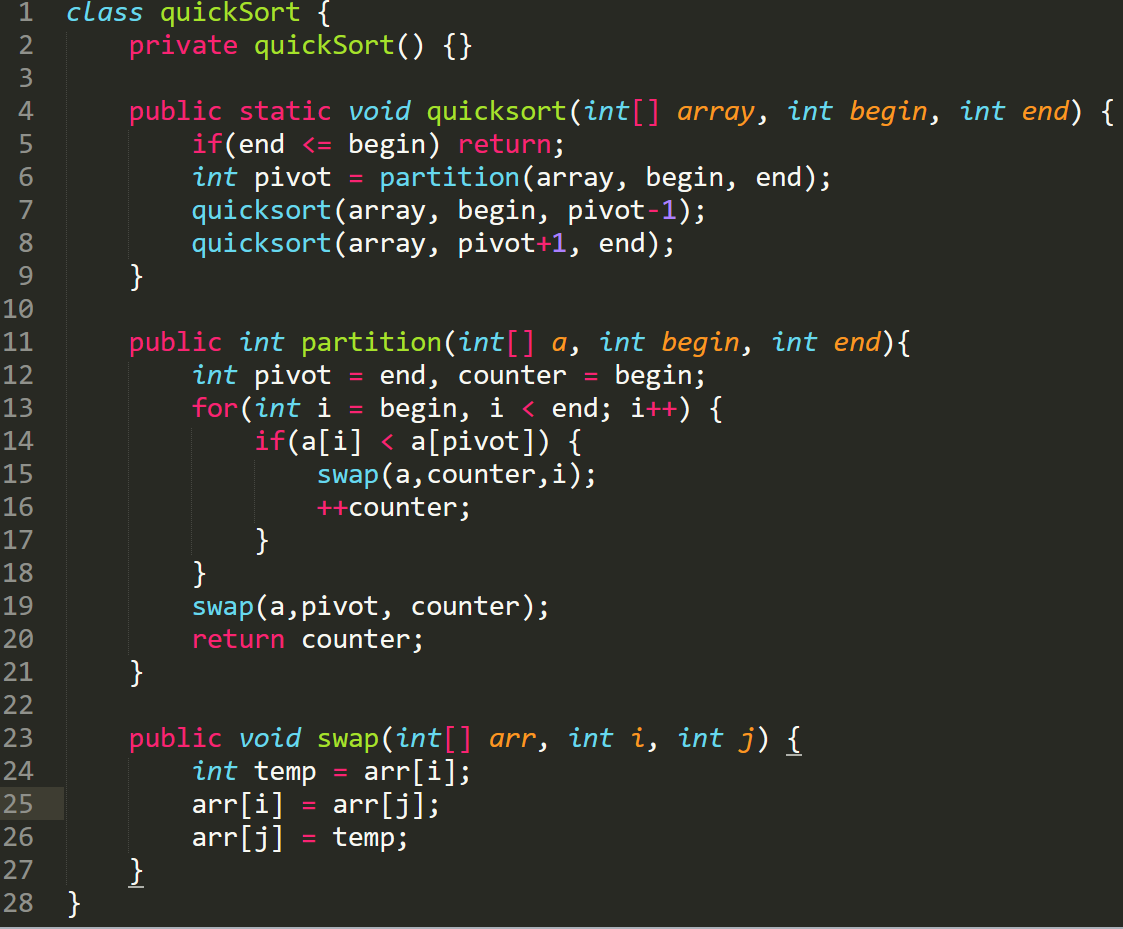


冒泡排序：

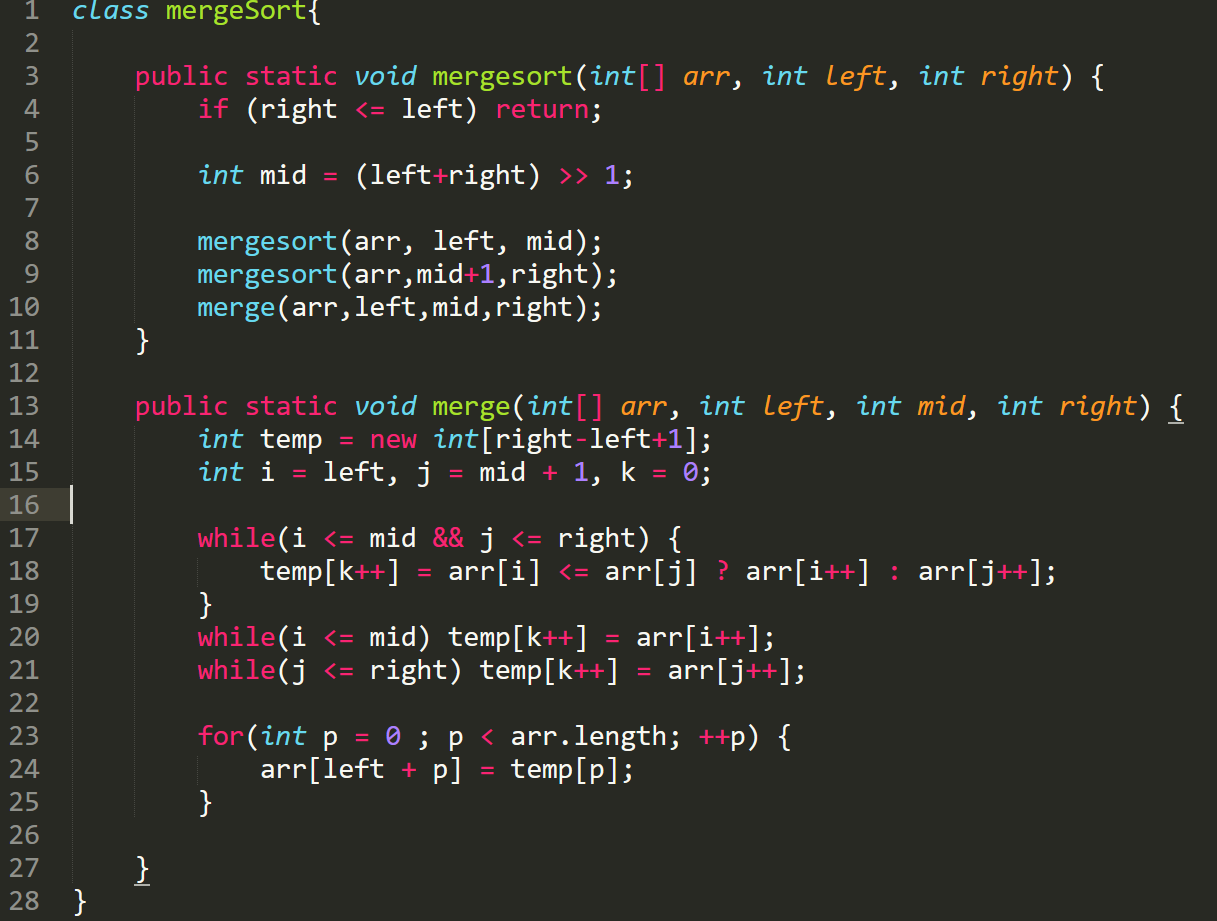


顺带也附上实现的归并排序和快速排序：

快速排序：



归并排序：



值得注意的是不同排序的亮点：

快速排序：使用了分治的思想，代码中最重要的是pivot和counter，先对传入的数组进行排序，然后得到pivot，此时能够保证pivot左边的变量都小于它所在的下标处的变量，而右边都大于。但是很明显可以看出它的缺陷。如果数值本身分布很极端，比如最小的数在end的位置，那么这样排序的时间复杂度是能够退化到O(n^2)级别的，换句话说，快排不是稳定的排序方法。

与快排相反，归并排序总是先进行二分，然后将左边和右边分别进行排序，排好序之后再将左右两边进行合并。由此一来，每次分割的数量总是确定的，所以归并排序是稳定的排序方法。

本周学习的内容除了排序算法，基本都是之前内容的加深。

让我印象深刻的，尤其是n皇后问题登峰造极的解法。之前一直以为超哥发出来的python的解法就是最好的了，现在才知道原来用位运算可以十行左右就搞定。

对于338题的解法，使用p(x)=p(x&(x-1))+1这个递推方程可以有很高的解题效率。

LRU cache的考题基本只会有一种，就是手动实现LRU cache。面试的时候最好还是可以手写双向链表+哈希表实现的方法。

排序算法经常会考，但是我很同意超哥的看法，就是初级排序算法（冒泡排序、选择排序、插入排序）因为时间复杂度平均是O(n^2)级别的，所以在实际工程中基本不用。而用得比较多的，也是面试常考的，是三个高级排序算法：快排、堆排序、归并排序（平均的时间复杂度都是O(nlogn)）,但是因为归并排序是稳定的，它用的空间复杂度也是最高的，可以算作是用空间换时间了。