本次作业我选择实现的是快速排序算法。因为要求图形化显示排序的步骤，并且生成视频，所以我选择了python调用OpenCV实现的方案。

快速排序和堆排序类似，但是在确定基准数值basic以后，从目标数组的两端交替查找，直到将将数组分成比basic大（或等于）和basic小的两部分，将basic插到中心再对这两部分分别排序。这一部分知识我们在本科的课程中曾经学过，而各类工具书中也反复介绍，所以我对快速排序算法的印象比较深刻，实现起来很轻松。遇到的唯一问题是：我习惯以排序区间的第一个值为basic，排序的过程中，小于basic的元素移到左边，大于等于basic的元素移到右边；当排序中的i和j相交的时候，将交点与basic的值交换，从而出现了问题。后来发现，将交点的值与basic交换，在basic的值等于交点的时候，会使左侧出现不小于basic的值，从而导致出错。所以我在最终实现的时候，都是将basic的值与交点的前一个值交换。

除了实现排序算法，我还着重设计了图形化显示和生成视频的部分。本来python有好多模块是专门画图用的。但是这次要求不同颜色和相应长度的柱形图，并且记录成视频，所以我专门使用OpenCV封装了一个方法：显示的柱形可以根据元素的数量调整宽度，根据最大元素的绝对值调整所有柱形的高度，并且每个元素都绑定了一个随机生成的颜色。在一个固定大小的画面上，随着元素位置的改变，相应柱形的高度和颜色是不变的。其次，在排序中参与排序的区间会用红框圈出，排序中的参考量basic、i和j也会被绘制在相应柱形的下方，充分反映排序的过程。OpenCV绘制完毕图像后，可以将图像展示出来，并且记录成avi格式的视频；所以每次运行程序，即使是同一个数组，每个元素对应的颜色也是不同的，并且都会生成一个新的结果视频文件。

运行程序前随意定义一个任意长度的数组，每当basic、i和j的位置变化，或者两个元素交换位置的时候，都会生成一张画面展示出来；而每次元素位置交换，命令行会打印出来新数组。程序执行完毕，会将结果展示5秒并生成一个每秒10帧的视频。

这次作业我一共写了一百多行代码，在不多的代码中，优化代码风格也是一项很重要的工作。这次作业中的函数名和变量名都是采用了驼峰式写法，从名称基本可以看出作用，并且进行了详细的注释，尤其是函数的参数都注明了作用，在后期可以直接放入其他工程中调用。