一，避免溢出

（1）将递归的程序改成非递归的程序，减少递归过程中的压栈占有的空间。

（2）用动态数组存放待排序的元素，避免不必要的开销。

（3）将待排序元素的数组作为全剧变量，避免将数组作为参数而引起的额外内存开销。

二，代码解释

我已经在代码中添加相应的注释。

第一问：

第一问对应的代码为problem1.cpp,我通过参考讲义上的5种排序算法的代码，编写了计算5种排序算法在0.5秒，1秒，5秒和10秒能够排序的个数。

在程序中，我将5种算法封装成函数，在main函数中调用这些排序算法。

首先，我通过c++中随机数种子产生随机数，以此构建待排序的数组。我将待排序的元素放在动态数组中，该动态数组作为全局变量。之后我通过调用c++中的clock\_t对象，计算各排序算法的运行时间。我将5种排序算法的函数均写在main函数中，当测试其中一种排序时，将其他排序的函数作为注释，只调用这种排序对应的函数。

第二问：

第二问对应的代码为problem2.cpp.与problem1.cpp一样，我将5种排序算法的代码封装成函数并在main函数中调用。

在main函数中，我设置了两重循环，第一重循环用于遍历5%M，10%M，50%M和M这四档，对应每一档设定不同的排序个数，先对待排序元素拍一次序并测试运行时间，再用第二重循环对待排序元素进行四次运行时间的测试。当五次运行时间测试完毕后，通过对五次的运行时间求和，将和除以五，得到平均值，再将计算得到的平均值作为参数之一传到函数caculateBiase（）中去计算标准差。

与第一问一样，当我测试某一个排序算法的运行时间时，我将其他排序的函数注释掉，只调用正在测试的排序算法的函数。

第三问：

在第二问的基础上，第一次测量时间之前利用插入排序（选择插入排序并没有什么理论依据）对待排序元素进行排序。

第四问：

因为第四问需要计算5组随机数据，并绘制折线图和柱状图，因此我利用c++的fstream库中的fstream类将运行时间和平均值、方差写到表格中，避免可复杂的手动操作，提高了效率

在将5种排序算法封装成函数后，我再main函数中通过三层循环实现第四问。第一重循环与二，三问相同，目的是遍历5%M，10%M，50%M和M，第二重循环是for循环，用于遍历5种排序算法，实现对于特定个数的待排序元素，分别用5种不同的排序算法进行排序。for循环内先对待排序元素进行一次排序并测试运行时间，再利用第三重循环进行后四次排序并测试运行时间。