实验 1：渐进分析和排序算法

本次实验内容包括两个方面的内容，一个是对课堂上所讲授的渐进分析加强认知和理解；一个是对若干个排序算法进行实践与比较。

实验要求：

1. 实验所用编程语言不限，可以自行选择如 Java、C、C++、python 等编程语言，能实现相应功能即可；
2. 实验报告撰写及提交要求：
   1) 实验报告需要同时具有封面和内容，封面排版按照封面模板文件要求执行，要具备目录；
   2) 实验报告命名为：姓名-学号，提交文件格式为 PDF，不需要提交源码，必要时可以在报告中插入代码或截图说明；
   3) 每一个实验任务的报告，应当从数据设计、算法设计、实现与测试等方面进行说明，可以适当根据题目要求进行变通；
   4) 任务中如果要求进行对比分析结果、总结收获和体会，务必要在实验报告的内容中有所体现；
   5) 实验中涉及到的绘图，推荐使用 JFreeChart 完成，也可以使用其他工具（如 Excel、matlab、Origin 等）完成图表绘制。

XTUSE

任务 1 证明

1) 使用 O、Ω和Θ的定义，证明下面每一个等式：
   a) $2\sqrt{n}+6=O(\sqrt{n})$
   b) $n^2=\Omega(n)$
   c) $log_2(n)=\Theta(\ln(n))$
   d) $4^n\neq O(2^n)$
2) 使用数学归纳法证明$T(n)=\Omega(nlog(n))$。T(n)的定义式如下：

$$T(n)=\begin{cases}2T\left(\left\lceil\frac{n}{2}\right\rceil\right)+\frac{n}{2}, & n>1\\ 1, & n=1\end{cases}$$

任务 2 排序算法的实现

请根据课上所讲，编程实现插入排序、选择排序、希尔排序、快速排序、归并排序。其中 Shell 排序中的间隔递减序列采用如下函数：

$$\begin{cases}h_1=1\\ h_i=h_{i-1}*3+1\end{cases}$$

要求：

- 每个排序算法使用课堂上所讲授的步骤，不要对任何排序算法进行额外的优化；
- 对每个排序算法执行排序之后的结果编写测试程序检查是否排序成功。

- 进行上述 5 种排序算法的对比分析，总结个人收获。

任务 3 排序算法性能测试和比较

完成对每一个排序算法在数据规模为：2^8、2^9、2^{10}、……、2^{16} 的均匀分布的随机数据序列、正序序列和逆序序列的排序时间统计。

要求：

- 在同等规模的数据量和数据分布相同下，要做 T 次（报告中请对 T 的取值进行说明）运行测试，用平均值做为此次测试的结果，用以排除因数据的不同和机器运行当前的状态等因素造成的干扰；
- 将所有排序算法的运行时间结果用图表的方式进行展示，X 轴代表数据规模，Y 轴代表运行时间。（如果因为算法之间运行时间差异过大而造成显示上的问题，可以通过将运行时间使用取对数的方式调整比例尺）
- 对实验的结果进行总结：从一个算法的运行时间变化趋势和数据规模的变化角度，从同样的数据规模和相同数据分布下不同算法的时间相对差异上等角度进行阐述。

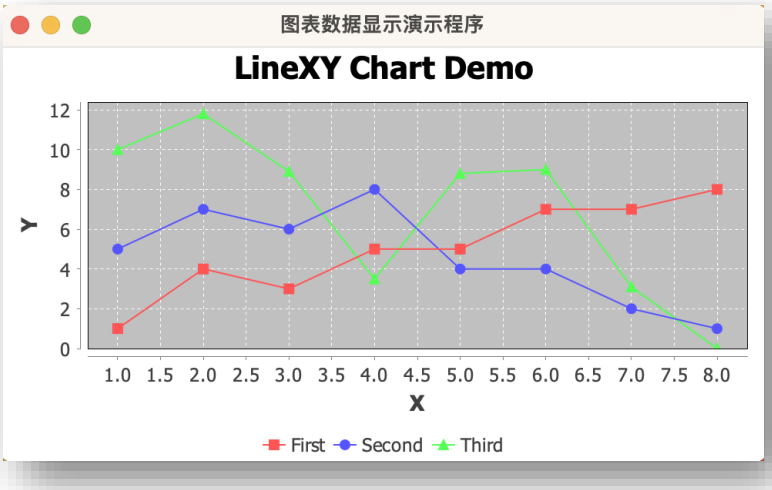
XJTUSE

补充材料：

实验任务需要将实验结果用图表的方式完成展示，完成这样的功能有很多方式，下面将介绍使用 Java 语言完成此功能的方式。

JFreeChart 是一个开源的 Java 语言编写的图表生成框架。支持生成饼图、条形图、折线图等相关图表。官方的网址：http://jfree.org/jfreechart/，在随实验的附件中还有两个文件：jfreechart.jar 和 LineXYDemo.java。其中 jfreechart.jar 文件是能够创建 JFreeChart 图表的类库，需要在工程项目中将该 jar 包加入到引用路径中，具体方式因开发工具的不同而有所不同，请自行查阅加入方式；LineXYDemo 是一个可以运行的使用 JFreeChart 框架下生成的线型图表，在源代码中有详细的注释供参考。

提供这些补充材料有两个目的：一是希望同学们更熟练地掌握Java面向对象编程语言的使用；二是希望同学们将主要精力放在数据结构的实践任务中（毕竟在 LineXYDemo 中只要简单替换其中的数据，就可以显示图表数据了）。有兴趣的同学，可以在课余时间继续了解 JFreeChart。下图是 LineXYDemo 运行的结果：



任务 4 快速排序的再探讨和应用

快速排序算法被誉为 20 世纪科学和工程领域的十大算法之一。前面的任务只是对快速排序的初识，下面从几个方面再更深入了解它：

① 优化快速排序：当待排序的数据量小于某个阈值时将递归的快速排序调用改为直接插入排序调用，按照这种策略的优化的快速排序算法进行实现、测试排序结果，并与任务 2 中没有优化的快速排序算法的执行效率进行对比；

② 在实际应用中经常会出现含有大量重复元素的数组，例如可能需要将大量人员资料按照生日排序，或者按照性别排序。给出使用①中完成的快速排序在数据规模为 2^{16} 的情况下，数据的重复率分别为 50%、60%、80%和 100%的运行时间的变化趋势图。结合①中的运行数据，给出观察结果；

③ 当遇到②中的重复性很高的数据序列时，快速排序还有很大的改进空间，方法是改进快速排序的划分方法，即将原有的二路划分（划分为比轴值大和不小于轴值）变成三路划分（划分为比轴值小，等于轴值和比轴值大）。三路划分的思路来自于经典的荷兰国旗问题。现要求自行查找三路划分的逻辑并实现之（高效的三路划分算法可以参考 J.Bentley 和 D.Mcllroy 的实现）。另外，用新的划分算法实现的快速排序重新对②完成实验并比较。

④ 在N个数据中找第k小元素(1≤k≤N)的问题可以有若干方法，请给出你能想到的方法，并简要分析每个方法的时间复杂度。在若干方法中，可以利用快速排序思想高效实现，请尽量独立思考，并最终给出设计思想、具体实现、测试以及时间复杂度分析。

XJTUSE