

第 17 课 选择排序轻松做

教学目标

1. 能够梳理出选择排序的基本步骤，通过补全流程图，了解用流程图描述的选择排序算法。
2. 了解排序数据量与算法执行次数的关系，能够用程序验证选择排序算法。

教学重点

1. 选择排序的基本过程。
2. 用流程图描述选择排序算法。

教学难点

能用流程图描述选择排序算法。

教学准备

具备上网条件的信息科技实验室、多媒体课件、视频等。

教学过程

新课导入

日常生活中，排序是很常见的事情，例如，把一堆水果按大小排序，把玩具球按直径大小排序，一队小朋友按身高排序，一个旅游团按年龄排序，一些动物按它们的奔跑速度排序……



【教师活动 1】你还知道哪些场景需要排序呢？

【学生活动 1】参与课堂学习活动，积极回答教师提出的问题。

【设计意图】引导学生快速回忆生活中排序问题，为本节课开展学习做好准备。

新知探究

一、描述选择排序的算法

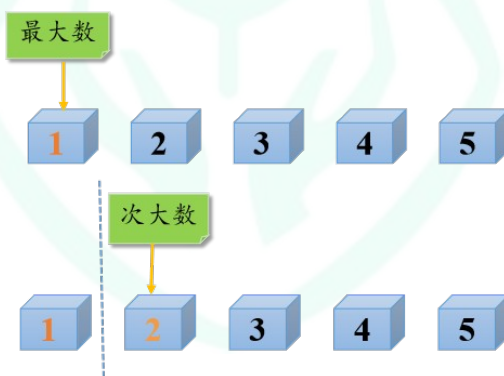
【教师活动 2】桌上有 5 个盒子，分别以 1、2、3、4、5 编号，每个盒子里装有一张卡片，卡片上写着一个大于 0 的数。请依据这些卡片上的数，把它们从大到小排序。

要求：写着最大数的卡片放在 1 号盒子，写着第二大数的卡片放在 2 号盒子，以此类推。



【学生活动 2】学生思考、讨论与交流自己排序的初步想法。

【教师活动 3】在排序算法中，如何将五个无序的数进行排序呢？如下图所示：



第 1 步：用前一课所学方法，找出最大数的卡片所在的盒子。

第 2 步：把写着最大数的卡片与 1 号盒子里的卡片进行交换。

第 3 步：以此类推，在 2 号到 5 号盒子里，用“找出最大数”的方法，继续查找其中最大数的卡片，找到后与对应盒子里的卡片进行交换。

【学生活动 3】学生聆听教师演示与讲解，思考排序过程。

【教师活动 4】以上操作，其实也形成了一个具有循环结构的算法。

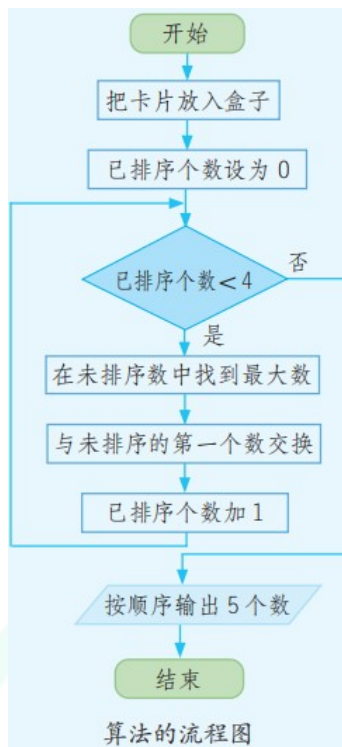
第 1 步：把卡片放入盒子，已排序个数设为 0。

第 2 步：在未排序数中找到最大数，与未排序的第一个数交换位置。

第 3 步：重复第 2 步，依次查看所有数并交换到相应的位置。

第 4 步：按顺序呈现 5 个数。

用流程图的方式表述这一算法如下图所示：

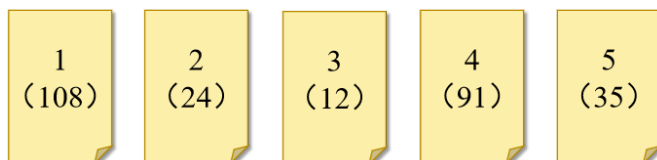


【学生活动 4】学生阅读此流程图，分析与理解排序算法的过程。

【设计意图】借助实物（5 个盒子、5 张卡片），通过查看、比较、记录盒子里卡片上的数，体验排序的步骤和基本过程。

二、分析选择排序的过程

【教师活动 5】这里的 5 张卡片上有 5 个数，如 108、24、12、91、35。运用前面所学选择排序算法，完成五个数从小到大的排序，验证算法执行的步骤，记录卡片交换的次数填写任务单。



【学生活动 5】学生填写任务单。



【教师活动 6】引导学生交流讨论，分析理解排序过程。

【教师小结】让学生将算法的执行过程填写到任务单中，进行展示和交流，可以检验他们对排序算法执行过程的了解程度。

【设计意图】通过直观的数值排序活动，让学生进一步巩固选择排序执行次数与数据顺序的关系，了解选择排序算法的执行效率。

三、验证选择排序的算法

【教师活动 7】在上一课的学习中，当找到列表中的最大数后，还可以进行多次比较和交换，实现列表中的数按从大到小排序。打开以下参考程序并运行，利用程序实现选择排序算法，把数据由大到小排序。

```
data = [64, 25, 12, 22, 11, 90, 88, 77, 55, 33] # 要排序的数据列表
n = len(data) # 获取列表数据的个数
i = 0 # 设置外层循环控制变量的初始值
while i < n - 1: # 外层循环用于控制排序的轮数
    max = i # 最大值变量设为当前轮数的起始值
    j = i + 1 # 设置内层循环控制变量的值
    while j < n: # 内层循环用于找到未排序数中的最大数
        if data[j] > data[max]: # 如果找到的值更大就更新最大数
            max = j
        j = j + 1
    if max != i: # 如果最大数不在指定位置就交换位置
        data[i], data[max] = data[max], data[i]
    i = i + 1 # 进入下一个数的排序
print('排序后的数:', data) # 输出排序后的列表
```

【学生活动 7】完成相关学习活动。

【教师小结】只有 5 个数时，人通过看一看、比一比，就能完成排序。但如果是 50 个数、500 个数、5 000 个数……就不能很快完成了。然而，通过算法编写程序让计算机执行，不管是 5 个数，还是 50 个、500 个、5 000 个……计算机都能够瞬间完成。

【设计意图】根据学生学习情况，引导学生依据算法流程图，分析程序语句的功能。通过利用程序对数据排序的验证，让学生进一步了解选择排序算法。

应用提升

编程验证

【教师活动 8】在 Python 中，可以使用 sorted() 函数来进行排序，编程验证与实现算法时更加简单便捷。打开配套资源中的参考程序，观察并运行程序。

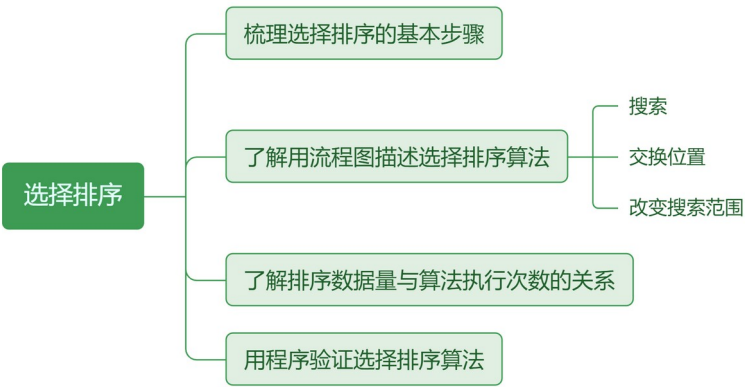
【学生活动 8】合作完成学习活动。

【设计意图】体会用程序实现算法的多样性，感受多途径解决问题的过程。

课堂小结

教师引导学生总结知识、分享收获。

- 1. 选择排序是计算机完成排序任务时常用的一种排序算法。
- 2. 选择排序算法借助在特定范围内，寻找最大数或最小数并交换位置的方式进行排序。
- 3. 通过探究选择排序与找出最大值或最小值之间的关系，可以了解算法的内在联系。



【设计意图】让学生通过梳理和总结本课知识点，巩固本节课所学知识的同时，对选择排序的算法有进一步的理解。

教学反思

