

信息技术

五年级 下册

主 编：李东涛

副主编：邱 华 罗柳垠 吴玮璇

清华大学出版社
北京

学 术 顾 问：雍俊海
总 主 编：刘 军
副 总 主 编：谢作如 夏正仁 陈 军
本 册 主 编：李东涛
本 册 副 主 编：邱 华 罗柳垠 吴玮璇

版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目 (C I P) 数据
信息科技 . 五年级 下册 / 李东涛主编 . -- 北京 :
清华大学出版社 , 2025. 1. -- ISBN 978-7-302-67676-8
I . G624.581
中国国家版本馆 CIP 数据核字第 2024UC2291 号

责任编辑：赵轶华
封面设计：王 静 薛玉斌 张思宇
责任校对：赵琳爽
责任印制：沈 露

出版发行：清华大学出版社
网 址：<https://www.tup.com.cn>，<https://www.wqxuetang.com>
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-83470000 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课程资源：<https://wqbook.wqxuetang.com/qhytl/>

印 装 者：新疆新华印务有限责任公司
经 销：全国新华书店
开 本：185mm × 260mm 印 张：5.75 字 数：106 千字
版 次：2025 年 1 月第 1 版 印 次：2025 年 1 月第 1 次印刷
定 价：18.80 元

产品编号：110774-02

前言



亲爱的同学们：

你们对计算机、手机和平板电脑里呈现的神奇世界感到好奇吗？那些精美的图文、有趣的视频和丰富的知识都是怎么来的？悄悄告诉你们，这一切都离不开信息科技！

在这个信息爆炸的时代，信息科技如同“魔法”般改变着我们的世界，它让学习变得更加便捷，也让生活变得更加多彩。从早上起床的手表提醒，到上学路上的导航指引；从课堂上的“双师”教学，到放学后的在线学习；从周末与家人的视频通话，到节日里用电子贺卡传递祝福……信息科技的身影无处不在。

作为互联网时代的“原住民”，同学们需要掌握一定的信息科技知识，才能更好地适应这个时代的快速发展。因此，我们精心编写了这套《信息科技》教材，希望它能够成为你们探索信息世界的“指南针”，带领你们走进信息科技的殿堂，解锁生活中的无限可能，发现更多令人惊喜的奥秘。

这套充满乐趣、智慧和挑战的“魔法”之书，将带领你们在探索中发现，在发现中成长。无论你是对互联



网感兴趣的小小网民，还是对编程充满好奇的小小程序员，或是想探索虚拟世界的小小探险家，这里都能满足你的需求。你们将像“探险家”一样，跟随我们的脚步，踏上一场充满乐趣和惊喜的“冒险”之旅。

本册教材包括查找与排序、经典算法设计、人工智能与算法三个单元，通过学习，你们将了解用自然语言、流程图或图形化编程的方式描述算法，知道一个问题有多种解决方式，不同的方式在时间效率上可能存在差别；学会将复杂问题拆解为子问题，通过解决子问题实现问题的解决；了解人工智能中的一些算法，体验机器识别、生成式人工智能的简单应用，理解可控技术与数据安全的重要性。

信息科技并不只是冷冰冰的计算机和代码，它更是一门充满活力和创造力的学科。学习信息科技，是一件既有趣又有意义的事情。在这本书中，你们或许会遇到一些困难和挑战，但只要保持好奇心和求知欲，勇敢地面对，就一定能够找到解决问题的方法，收获满满的成就感。

亲爱的同学们，让我们一起开始这段奇妙的旅程吧！愿你们在学习中获得知识、收获快乐、收获成长！在学习的过程中，你们一定会成为掌握信息科技的小小“魔法师”，用你们的智慧和勇气创造出更加美好的明天！

——编委会的大朋友们

目 录



1 查找与排序 1

第 1 课	顺序查找.....	3
第 2 课	二分查找.....	9
第 3 课	数据排序.....	16
第 4 课	活动：算法效率对比.....	23

2 经典算法设计 31

第 1 课	枚举算法.....	33
第 2 课	递推算法.....	40
第 3 课	递归算法.....	46
第 4 课	跨学科活动：斐波那契数列 与黄金比例.....	51

3 人工智能与算法 59

第 1 课	智能推荐.....	61
第 2 课	智能路径规划.....	67
第 3 课	智能识别.....	74
第 4 课	活动：体验生成式人工智能.....	80

清华大学出版社

1

查找与排序

📖 本单元你将学习：

- 顺序查找的算法思想以及顺序查找的优缺点
- 二分查找的算法、分治的原理
- 如何利用二分查找解决实际问题
- 冒泡排序的原理方法以及插入排序算法的基本思想
- 如何比较两种算法解决相同问题的效率

在计算机的世界里，查找和排序是我们解决问题的基础工具之一。想象一下，当你在图书馆找一本书时，你需要使用合适的方法快速找到它，而排序就像是把书架上的书按编号有序排列，让你轻松地根据编号找到需要的那本书。

本单元，我们将一起探索顺序查找和二分查找的奥秘，了解它们如何帮助我们高效地寻找信息。同时，我们还会学习不同的排序方法，让我们的数据更加整齐有序。希望你能在这个过程中发现乐趣，激发对计算机科学的热爱！



学习热身

同学们，你们有没有在家里找书、文具或衣服的经历呢？分享一下你找到这些东西的经历吧！

- 哪一次你找到物品的速度最快？
- 又是哪一次浪费了很长时间呢？
- 花费时间的多少与哪些因素有关？

正确的寻找方法会为你节约大量的时间，接下来就让我们一起来研究查找与排序算法吧。

遍历 排序 链表
顺序查找
二分查找
分治原理
算法效率



你知道吗？

信息设备中存储的内容均为数据，在处理这些数据时，查找与排序的操作极为常见，采用正确且高效的算法能够显著提升程序的运行效率。



第1课

顺序查找



本课中你将学习：

- 顺序查找的工作原理
- 顺序查找的特点
- 利用顺序查找法找到目标

查找是最常用的算法之一，是指通过一定的方法找出与给定关键字相同的数据元素的过程。

顺序查找也称为线性查找，是一种简单且常用的查找方法。在我们的日常生活中，比如依次查阅书架上的书籍、挨个翻找衣柜里的衣物、逐个在操场上寻找同学，都无形中运用了顺序查找。掌握这一基础的查找算法，将为学习其他更复杂的查找算法奠定坚实的基础。

一、顺序查找的原理

学校要组建国旗队，需要在小青班里挑选一位身高等于 148cm 的同学作为学校的国旗手，小青需要根据全班同学体检报告中的身高数据，找出符合要求的同学。

面对复杂的问题，我们可以通过缩小问题规模去探索问题的本质，从而找到解决问题的方法。小青首先以自己所在的 7 人小组为例，尝试查找符合身高要求的同学，7 位同学的体检表如图 1.1.1 所示。

1 体检表	2 体检表	3 体检表	4 体检表	5 体检表	6 体检表	7 体检表
----------	----------	----------	----------	----------	----------	----------

图 1.1.1 7 位同学的体检表



体检表上的编号 1、2、3、…、7 被称为序号，它是数据项目的编号，与数据值的大小没有直接关系。

第 1 步：翻开第一份体检表，第一位同学的身高为 143cm，不符合要找的 148cm（图 1.1.2），所以继续翻开下一份体检表。

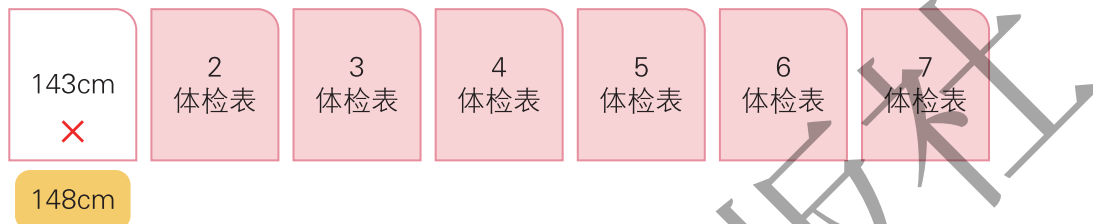


图 1.1.2 体检表查找第 1 步

第 2 步：翻开左起第二份体检表，这位同学的身高为 149cm，也不符合要求（图 1.1.3），所以继续翻开下一份体检表。

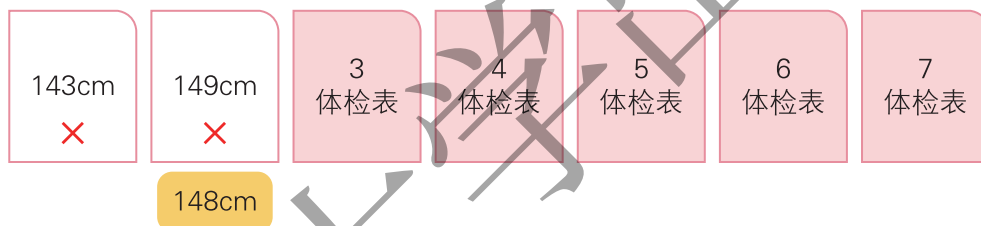


图 1.1.3 体检表查找第 2 步

第 3 步：重复上面的操作，直到翻开第 5 张体检表，第 5 位同学的身高为 148cm，符合要求，完成查找（图 1.1.4）。

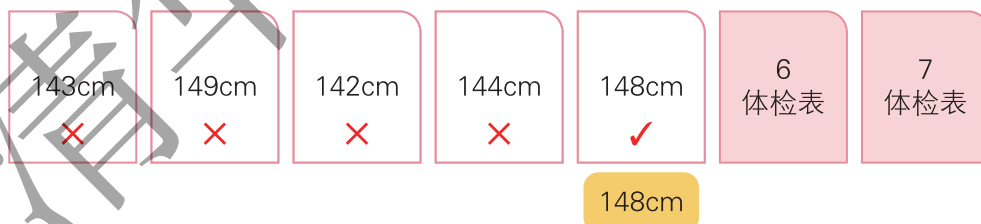


图 1.1.4 体检表查找第 3 步

通过上面的例子，我们可以知道，顺序查找的原理就是按照数据序列原有的顺序，从第一个元素开始逐个查找，直到找到与给定关键字相同的元素或者搜索完整个数据序列。

这种按顺序逐个访问集合中所有元素的过程也称为遍历。在编程和数据处理中，遍历是一种基础且重要的操作。在日常生活中，老师课前按学号顺序逐个点名，体育课时挨个报数也是一种遍历。

探索

在查找的过程中，你们发现了顺序查找的哪些优缺点，填写在表 1.1.1 中。

表 1.1.1 顺序查找的优缺点

优点	缺点

二、顺序查找的最大查找次数

在顺序查找中，如果要查找的元素刚好是数据集中的第一个元素，那么查找次数为 1，这是查找次数最少的情况。如果要查找的元素位于数据集的最后一个位置，或者数据集中根本不存在该元素（此时算法会遍历整个集合以确定这一点），那么查找次数将是最多的。总之，查找的次数不会超过样本总数。



我们班一共有 40 份体检表，我使用顺序查找，在最坏的情况下，我要找多少次才能找到旗手呢？

最大查找次数不可能超过 40 次。





探索

想一想，将正确答案填在下面的括号里。

1. 如果有 100 份样本，最大查找次数是多少次呢？



2. 如果有 n 份样本呢？



三、用顺序查找法查找旗手

1. 使用自然语言、流程图描述查找旗手的过程

首先，我们用最简单的方法——自然语言，来描述查找旗手的过程：从第一份体检表开始，取出身高数据，与目标身高进行对比，如果不相等，就依次向后查找，直到查找到目标旗手为止。

请你使用流程图描述用顺序查找法查找旗手的步骤，在图 1.1.5 中填写空白部分。

2. 编程查找旗手

下面用图形化编程软件编写程序，根据全班同学的身高数据找出身高为

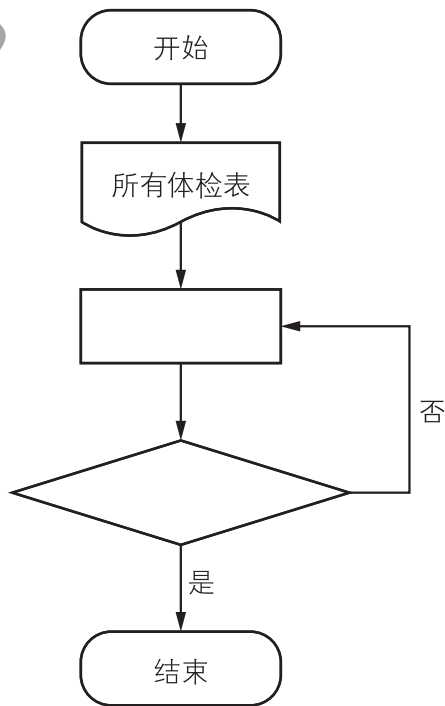


图 1.1.5 顺序查找流程图

148cm 的旗手。

(1) 保存全班同学的身高数据。



老师，一位同学的身高数据可以用变量来存储。那 40 位同学的身高数据难道要新建 40 个变量吗？有更好的办法吗？

我们可以用图形化编程软件中的“列表”来存放多个数据，还可以很方便地对列表数据进行插入、删除、替换、复制等操作。



在图形化编程软件中，新建名为“全班身高”的列表，如图 1.1.6 所示。将文本文件中的列表数据复制并粘贴到“全班身高”列表中，如图 1.1.7 所示。



图 1.1.6 新建“全班身高”列表



图 1.1.7 复制文本文件中的数据并粘贴到列表中

(2) 遍历“全班身高”列表数据。

尝试通过“重复执行 () 次”积木，遍历列表数据，注意重复执行的次数要等于列表的长度，如图 1.1.8 所示。



图 1.1.8 遍历全班身高数据

探索

说一说变量 n 在上面代码中的作用。

挑战

上面的代码将列表内的数据从头到尾进行了遍历，你能修改代码，让其实现当判断到列表中某个内容等于 148 时说出数据的编号并停止执行吗？

第2课

二分查找



本课中你将学习：

- 二分查找的工作原理
- 利用二分查找找到目标
- 二分查找在生活中的应用

上节课，我们学习了顺序查找，小清提出了新的疑惑，假如有 1000 个数据，难道我们要查找 1000 次吗？有没有更快的查找方法？本节课，就让我们来学习一种效率更高的查找方法——二分查找法。

一、二分查找的原理

当老师让同学们把书翻到第 50 页时，大家都采用什么方法翻书呢？

我是根据书的厚度，估计着翻一下，然后看一下页码，如果翻过了就往回再翻一下，反复几次就翻到了。

我明白了，从头一页一页翻，就像是顺序查找。

有没有人从头一页一页翻的？

从中间翻书难道就是二分查找？





在翻书时，书本的页码总是按顺序从小到大排列。我们可以通过不断翻到中间的页码，然后与目标页码进行对比，缩小查找范围，重复以上的操作，最终找到目标。

比如总页数为 120 页的书籍，要翻到第 50 页，我们可以先大概翻到中间的位置，如第 60 页左右；发现比第 50 大，于是再往前翻，翻到第 30 页左右，再依次大概翻到第 45 页、第 52 页、第 48 页，最后成功翻到第 50 页。

同理，在有序数列中，要想找到某个特定目标元素，可以通过将查找范围缩小一半，判断目标元素所在的位置，将目标元素所在的范围再次缩小一半，判断目标元素所在的位置……如此重复不断地缩小查找范围，直到找到目标元素。这样的方法就叫作二分查找。

探索

想一想，如果是一本没按顺序整理的资料，要找到某一页，可不可以用二分查找？

二、用二分查找法查找旗手

在用顺序查找法查找旗手的过程中，如果老师给我们的数据是按照从小到大或者从大到小的顺序排列的，我们就可以用二分查找法来简化问题，从而实现快速查找。

为了方便演示，我们以 10 份排好顺序的数据为例，查找身高为 148 cm 的同学。

第 1 步：将 10 份数据分为两部分，翻开 5 号体检表，发现其值为 144，小于 148，因此查找范围缩小至 6 号至 10 号体检表，如图 1.2.1 所示。

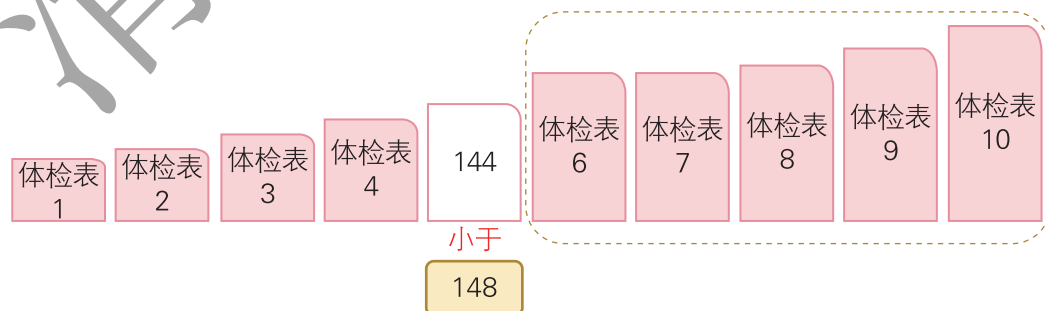


图 1.2.1 5 号体检表数据小于目标数值

第2步：将6号至10号体检表再次分为两个部分，翻开8号体检表，发现其值为147，仍然小于148。查找范围再次缩小至9号至10号体检表，如图1.2.2所示。

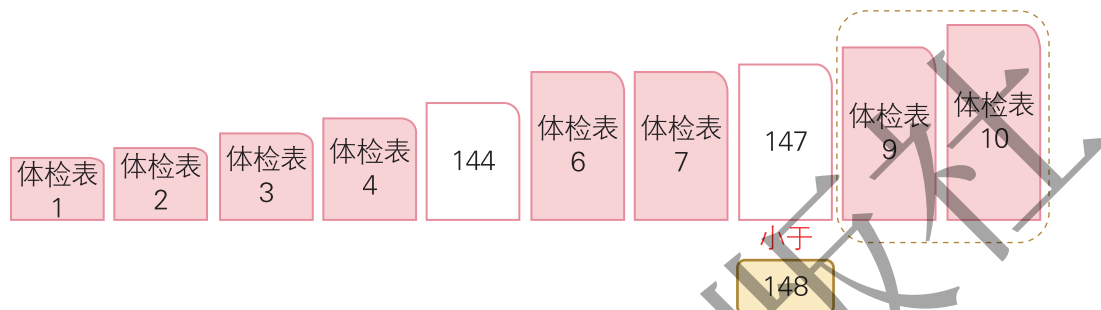


图 1.2.2 8号体检表数据小于目标数据

第3步：还剩下2份体检表，翻开9号体检表，其值等于目标数据148，查找结束，9号体检表中的身高符合旗手的身高要求，如图1.2.3所示。

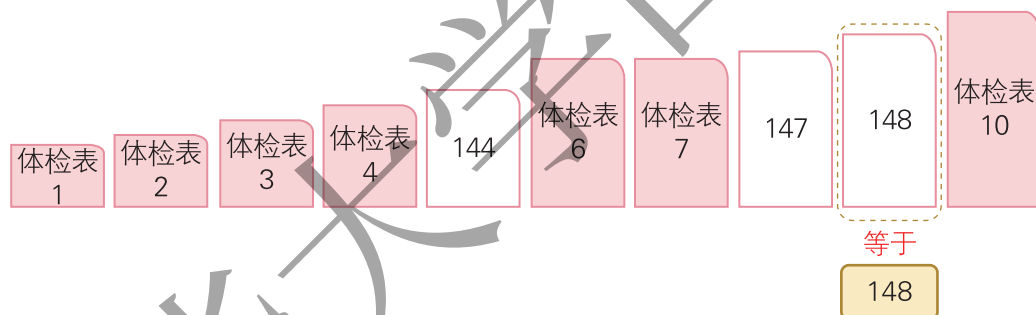


图 1.2.3 9号体检表数据等于目标数据

二分查找的原理就是在有序数组中，通过不断将查找区间一分为二，比较中间元素与目标数据，再根据比较结果调整查找区间，直至找到目标数据。

二分查找的最坏情况是每次查找都未能找到目标数据，每次将搜索范围缩小一半，直到搜索范围为空或只剩下一个元素时才停止。如果有 n 个元素，每次查找的元素数量除以2，直到元素数量等于1，查找结束。

阅读

如图1.2.4所示，要想将一条纸带均匀分为64个小格子，只需连续对折6次，展开后即可得到64个大小相等的小格子。

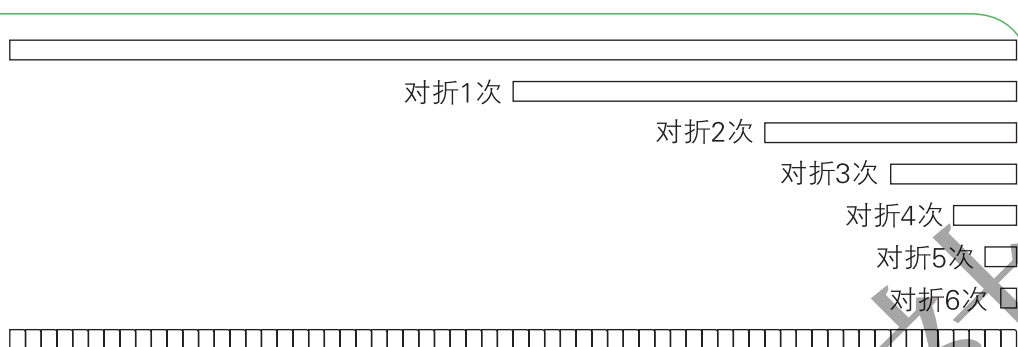
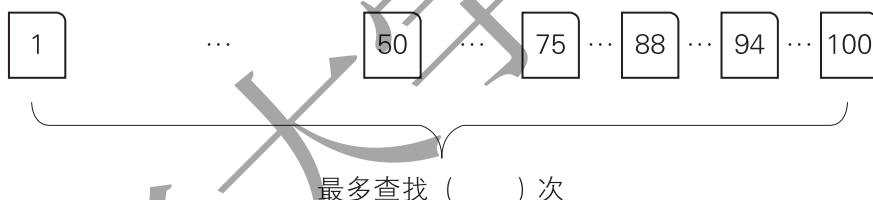


图 1.2.4 对折纸带

像二分查找、折纸带这样，将复杂的问题分解为较小的子问题，再分别解决这些子问题，最后将子问题合并解决原问题的策略也叫作分治法。

探索

使用二分查找，在有 100 个样本的情况下，最多需要查找多少次呢？



你知道吗？

当有 10 万亿个样本时，在最坏的情况下，二分查找只需要 43 次；而顺序查找在最坏的情况下则需要 10 000 000 000 000 次。

对比二分查找与顺序查找，我们发现，在数据有序时更适合 _____ 查找，在数据无序时更适合 _____ 查找。



三、二分查找的应用

在生活中，我们常常运用二分查找法的原理解决问题。比如，按照拼音查字典、猜数字、排除故障等。

如图 1.2.5 是一幅电路图。7 个开关触点可以用夹子夹住，测试电路连通情况。当黄色线路连通，灯泡会被点亮。现在电线 1 到 8 段中的某一段出现了故障，小青用二分查找的原理迅速找到了故障位置。他的方法是这样的：第一次连接中间触点 D，发现灯泡不亮，于是判定故障一定在 1 至 4 段；第二次连接触点 B，发现灯泡亮起，说明 1、2 段正常，故障范围缩小为 3、4 段；第三次连接触点 C，发现灯泡不亮，得出结论：故障在 3 段。

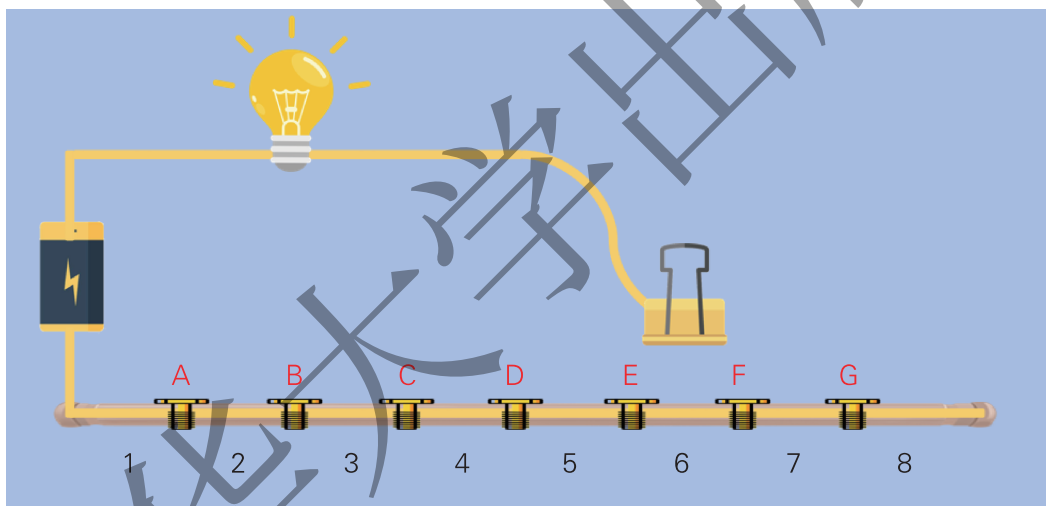


图 1.2.5 用二分查找法判断故障

而在猜数字游戏中，如果要猜中 100 以内的一个整数的概率是百分之一，假如利用二分查找的原理来猜，猜中的概率就会大幅提升。

猜数字游戏小程序如图 1.2.6 所示，具体规则是：生成一个 1 至 100 的随机数，当猜数字的同学输入答案后，系统会自动判断输入的数是否正确，不正确的话是大了，还是小了，并给出相应的提示。参与者根据提示继续猜，直到猜对为止。和你的同学一起试试吧！



图 1.2.6 猜数字游戏

探索

请根据猜数字游戏规则，思考并填写图 1.2.7 所示流程中空白的内容。

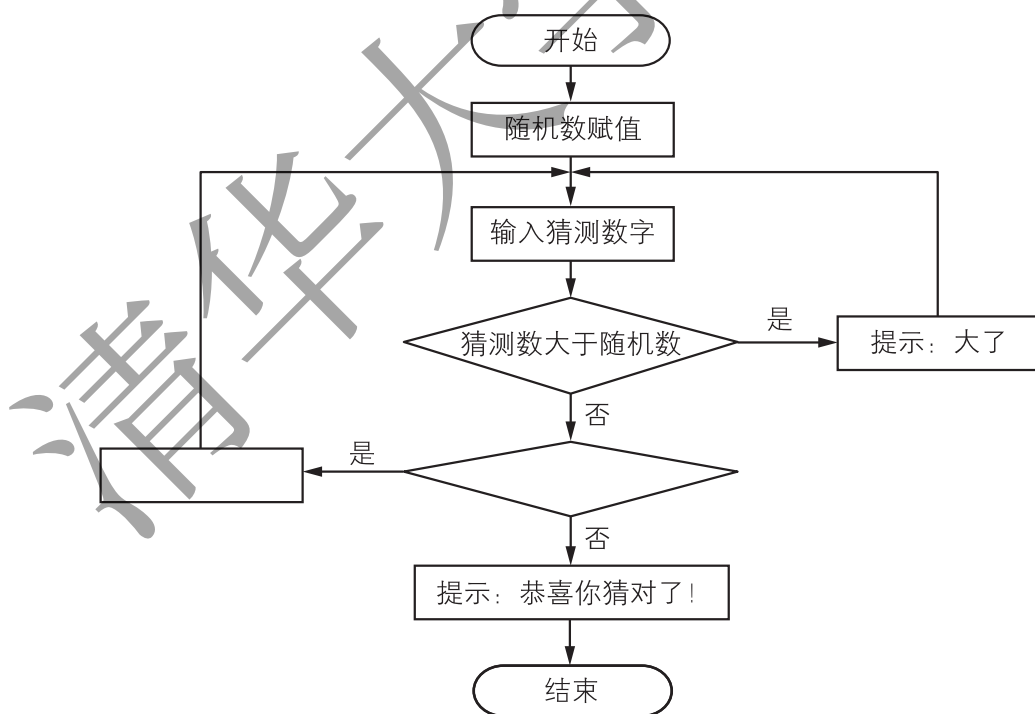


图 1.2.7 猜数字游戏流程

挑战

请打开图形化编程软件，参照图 1.2.8，用二分查找原理编程实现猜数字游戏吧！



图 1.2.8 用二分法编程实现猜数字游戏

第3课

数据排序



本课中你将学习：

- 冒泡排序的工作原理
- 冒泡算法过程
- 插入排序的原理

小青觉得，如果能将所有同学的体检表按照身高有序排列，那么以后无论需要寻找身高为多少的同学，都能快速找到。我们有什么好办法将体检表有序排列吗？有的同学想到，在集体活动站队时，大家先随便排成一排，然后通过换位置的方式逐步形成高矮顺序。其实这就是我们今天要学习的数据排序。

一、冒泡排序的原理



你知道吗？

为什么水沸腾时，下面的气泡小，上面的气泡大呢？其实是因为水面下的压力比水面上大。当水的温度达到沸点时，水面下的气泡就会形成并上升到水面，由于气泡在上升过程中受到的压力越来越小，气泡就会逐渐变大，如图 1.3.1 所示。



图 1.3.1 水沸腾时会“冒泡”

冒泡排序的基本思想是通过比较相邻两个元素，根据比较结果和算法规则，对这两个元素的位置进行交换。这样依次进行比较和交换，就能达到排序的目的。

由于这种方法在排序的过程中总是把小的数往下放，大的数往上放，就像水沸腾时气泡往上升会变大一样，所以叫作冒泡排序。

现在，有6个从下往上无序排列的数据泡泡：“12，1，27，19，30，8”。让我们来具体看看是怎样通过交换位置进行冒泡排序的吧。

第一轮排序如下。

第1次排序：将12与1比较，因为 $12 > 1$ ，所以交换位置。

第2次排序：将12与27比较，因为 $12 < 27$ ，所以不交换位置。

第3次排序：将27与19比较，因为 $27 > 19$ ，所以交换位置。

第4次排序：将27与30比较，因为 $27 < 30$ ，所以不交换位置。

第5次排序：将30与8比较，因为 $30 > 8$ ，所以交换位置。

至此，第一轮排序一共比较了5次，将最大的数字30交换到了最上方，得到的从下往上的序列为“1，12，19，27，8，30”。排序过程如图1.3.2所示。

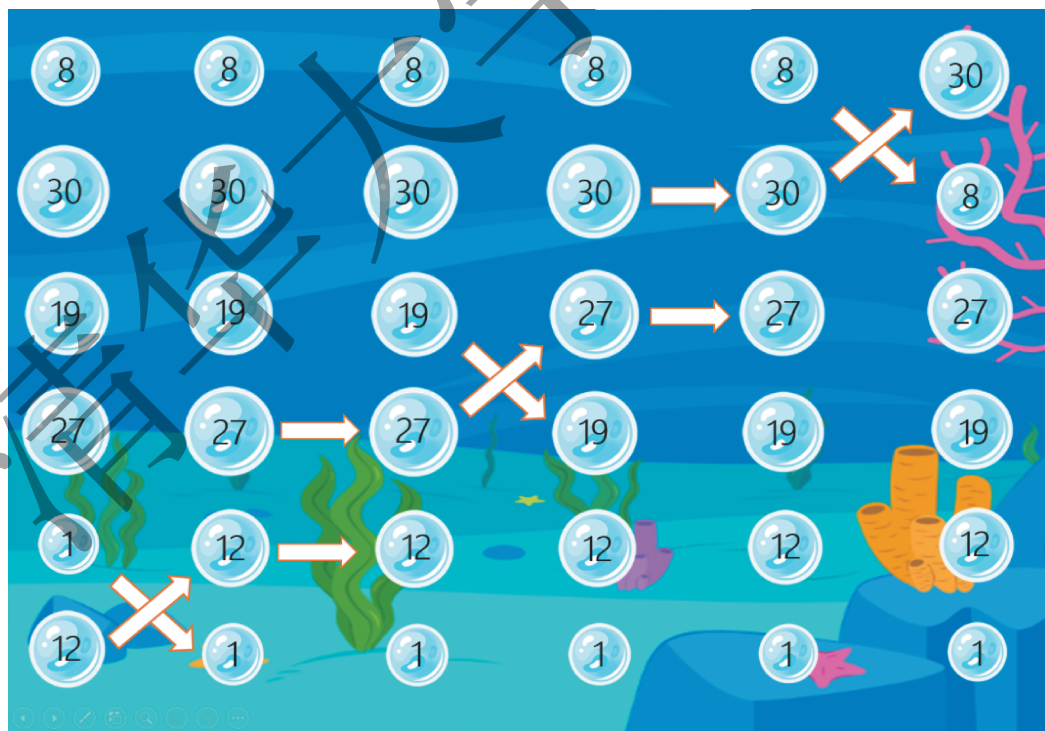


图 1.3.2 第一轮排序



探索

你能按照上述的思路和方法，在图 1.3.3 中的泡泡里填写对应的数字，完成第二轮的排序吗？



图 1.3.3 第二轮排序

二、冒泡排序算法过程

在每一轮排序后，都记录下最后一次元素交换的位置（表 1.3.1 中每一轮结果中蓝色的数字），那个位置就是无序序列的边界，再往上就是不再参与排序的有序序列了，所以每一轮会减少一个比较元素。

以此类推，我们就可以得出每一轮冒泡排序的结果，直到从下到上的数据都按照从小到大的顺序排列，就完成了排序，如表 1.3.1 所示。看一看是不是和你想的一样呢？

表 1.3.1 冒泡排序过程

轮次	比较次数	本轮结果
第一轮	5	1, 12, 19, 27, 8, 30
第二轮	4	1, 12, 19, 8, 27, 30
第三轮	3	1, 12, 8, 19, 27, 30
第四轮	2	1, 8, 12, 19, 27, 30
第五轮	1	1, 8, 12, 19, 27, 30

探索

请根据前面活动中冒泡排序的过程，思考并填写图 1.3.4 所示流程中空白的内容。

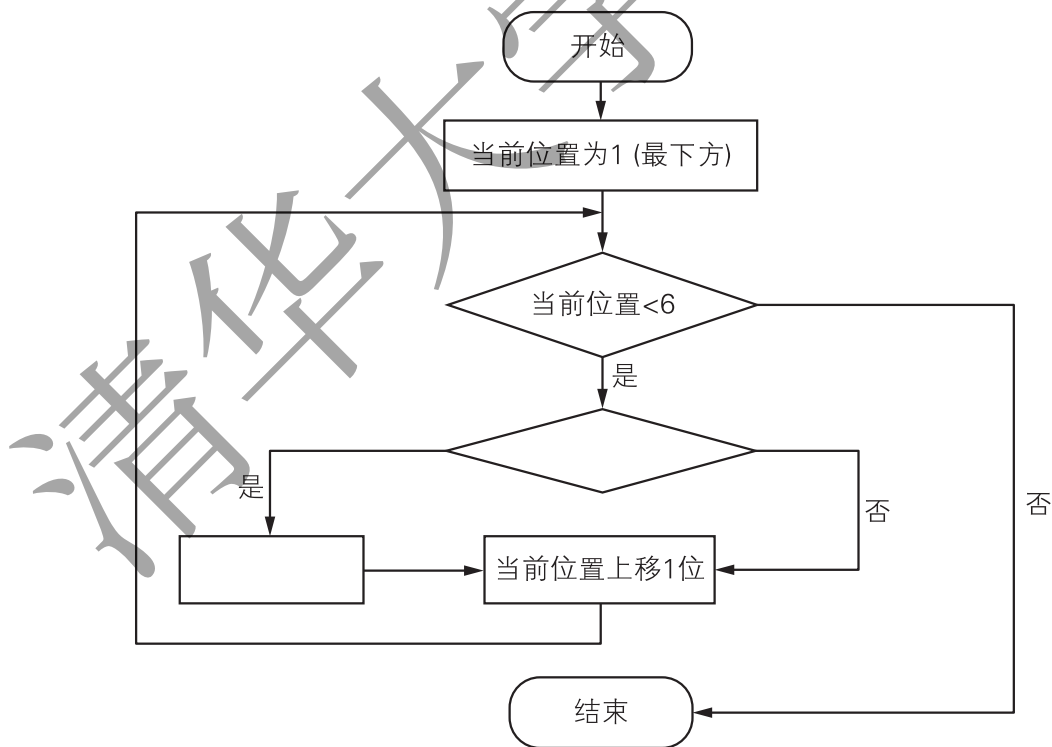


图 1.3.4 冒泡排序流程



三、插入排序的原理

插入排序是另一种较为常见的排序方法，我们在卡牌游戏中常常会用到这种方法。每次抽牌后都将新抽到的牌与手中的牌进行比较，按照大小顺序插入对应的位置。

插入排序的思想是将待排序的元素序列看作是一个有序序列和一个无序序列的组合。初始时，有序序列仅包含一个元素，即第一个元素，其余为无序序列。然后，每次从无序序列中取出一个元素，将其按大小插入有序序列的适当位置，使之成为新的有序序列。这个过程重复进行，直到无序序列为空，整个序列变为有序。

我们以放在桌面上的 6 张乱序卡牌为例，如图 1.3.5 所示。

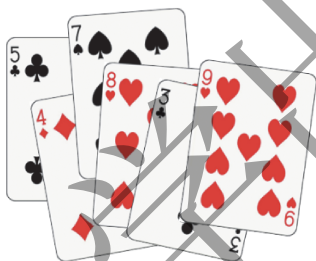


图 1.3.5 乱序卡牌

第 1 步：拿起卡牌 9，由于没有对比，所以直接放到右边空白处，如图 1.3.6 所示。

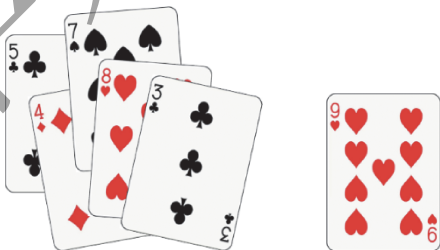


图 1.3.6 插入排序第 1 步

第 2 步：拿起卡牌 3， $3 < 9$ ，所以将 3 直接插入 9 的左边，如图 1.3.7 所示。

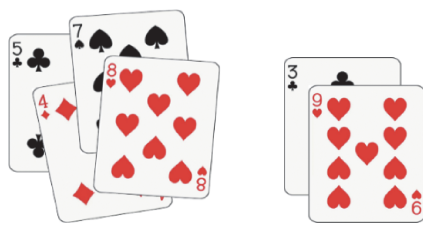


图 1.3.7 插入排序第 2 步

第 3 步：拿起卡牌 8， $8 > 3$ 且 $8 < 9$ ，所以将 8 直接插入 3 的右边、9 的左边，如图 1.3.8 所示。

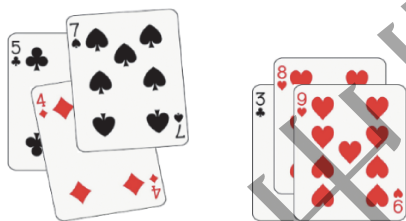


图 1.3.8 插入排序第 3 步

第 4 步：拿起卡牌 7， $7 > 3$ 且 $7 < 8$ ，所以将 7 直接插入 3 的右边、8 的左边，如图 1.3.9 所示。

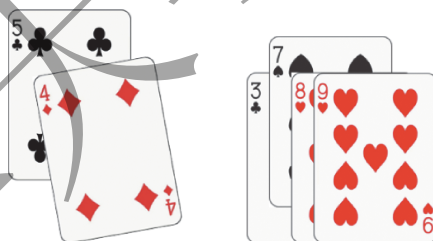


图 1.3.9 插入排序第 4 步

第 5 步：拿起卡牌 4， $4 > 3$ 且 $4 < 7$ ，所以将 4 直接插入 3 的右边、7 的左边，如图 1.3.10 所示。

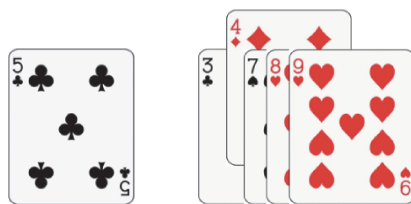


图 1.3.10 插入排序第 5 步



第6步：拿起卡牌5， $5 > 4$ 且 $5 < 7$ ，所以将5直接插入4的右边、7的左边，如图 1.3.11 所示。



图 1.3.11 插入排序第6步

至此，排序完成，数组变为有序数组，依次为“3，4，5，7，8，9”。

探索

冒泡排序与插入排序，哪种排序移动的次数较少呢？你更喜欢哪种方法？

挑战

常见的排序算法还有选择排序、桶排序、希尔排序和归并排序，请上网查询资料，向同学介绍它们的用法。

第 4 课

活动：算法效率对比



本课中你将体验：

- 从执行步骤比较算法的效率
- 从输入规模比较算法的效率
- 空间复杂度对算法效率的影响

伴随着互联网和移动设备的广泛普及，算法驱动的服务已经悄然融入了我们的学习、工作和生活中（图 1.4.1），算法的效率也因此变得越来越重要。你是否遇到过这样的情况：在手机或计算机上操作一个 App 时，执行某个指令需要等待很久，或者在玩游戏时，画面突然卡顿？这些情况往往可能是由于背后的算法效率不佳所导致的。



图 1.4.1 互联网时代



读一读

运行任何一个算法都需要占用时间与空间，时间是指算法执行完毕所消耗的时间，空间是指对计算机存储设备的占用。

如何考查算法的性能或者执行速度呢？这就要引入时间复杂度的概念。简单地说，时间复杂度就是指输入数据大小为 n 时，算法运行所需要耗费的时间。我们一般考查的是在最差情况下算法的时间复杂度。描述时间复杂度一般用大“O”表示法。大“O”表示法不是具体表示代码的执行时间，而是指当算法处理的数据量变得无穷大时，算法的性能将趋近于一个什么样的值，即一个算法的增长速率，也就是在最坏的情况下算法执行所需要的时间。

算法执行的步骤与时间是影响算法效率的重要因素。接下来，我们将通过一个简单的实验，来比较顺序查找算法与二分查找算法在解决同一问题时表现，以揭示其中的奥秘。



议一议

如何比较两种算法的效率呢？

综合大家的意见，我们可以用图形化编程来直观测试算法效率，再将结果用数据可视化的方式展示出来。

我们可以先用图形化编程实现两种查找方法，再进行对比。

对比的时候，简单和复杂的数据都要有。

对比两种算法，看看哪种算法执行的步骤较少，哪种算法所用时间较少。



用顺序查找算法与二分查找算法分别编写一段程序，计算从 $0 \sim n$ 的自然数中查找目标数据所需的最大查询次数与所需的时间，并运行测试当 $n=10$, $n=100$ 或其他更大数值时，两种算法的最大查询次数与所需的时间。分组记录结果，将结果用电子表格进行统计，最终将数据可视化后形成结论，小组分享交流。



为了节约时间，我们小组决定由一部分同学搭建顺序查找，另一部分同学搭建二分查找。

1. 编程寻找顺序查找与二分查找的最大查询次数

程序中，将“ n ”与“执行次数”设为两个变量，以便后面调节 n 值，统计最大查询次数。

设置数据量 n 为 1000，计算顺序查找算法的最大查询次数和二分查找算法的最大查询次数，如图 1.4.2 和图 1.4.3 所示。

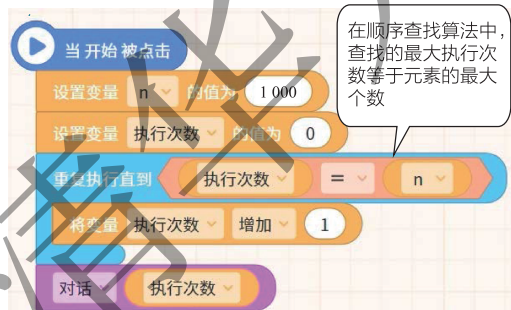


图 1.4.2 顺序查找算法的最大执行次数

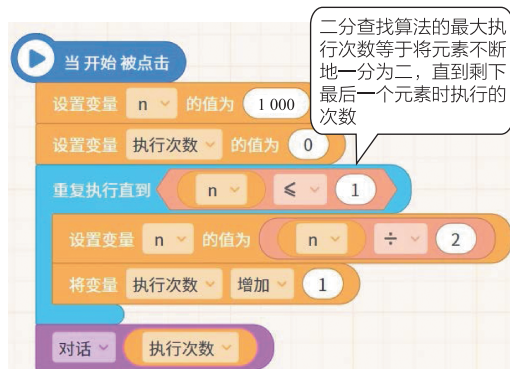


图 1.4.3 二分查找算法的最大执行次数

2. 为程序添加计时器，以便直观地获取程序运行的时间

在查询程序的开始与结束处添加上“计时器”模块，如图 1.4.4 和图 1.4.5 所示。



图 1.4.4 为顺序查找程序添加计时器



图 1.4.5 为二分查找程序添加计时器

3. 设计电子表格，统计实验结果

按照 n 的不同取值，在表 1.4.1 和表 1.4.2 中填写两种查找方法得出的最大执行次数与执行时间，其中首列为测试的算法名称，首行填写 n 的取值。

表 1.4.1 顺序查找与二分查找的最大执行次数

n 的取值	1	10	100	500	...
顺序查找的最大执行次数					
二分查找的最大执行次数					

表 1.4.2 顺序查找与二分查找的执行时间

n 的取值	1	10	100	500	...
顺序查找的执行时间					
二分查找的执行时间					

4. 实验数据可视化

将测试得出的数据做成相应的图表。想一想，实验得出的数据用什么形式的图表来表示比较合适呢？

柱状图与折线图各自都有自身的优点，柱状图能直观地比较数据的大小，折线图可以反映出数据变化的趋势。

使用表 1.4.1 中的最大执行次数和表 1.4.2 中的执行时间数据，分别生成柱状图和折线图，如图 1.4.6 和图 1.4.7 所示。

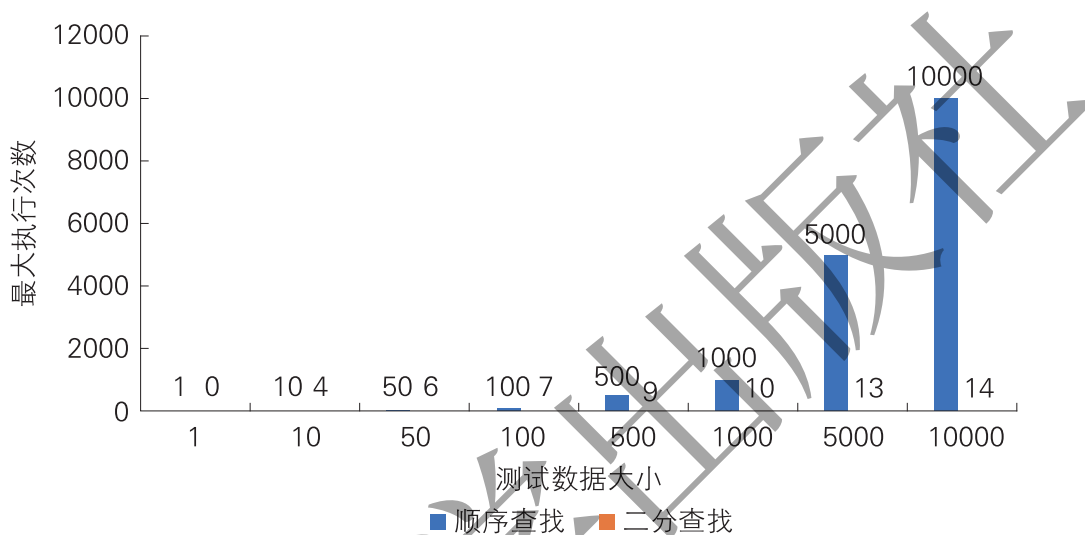


图 1.4.6 最大执行次数对比

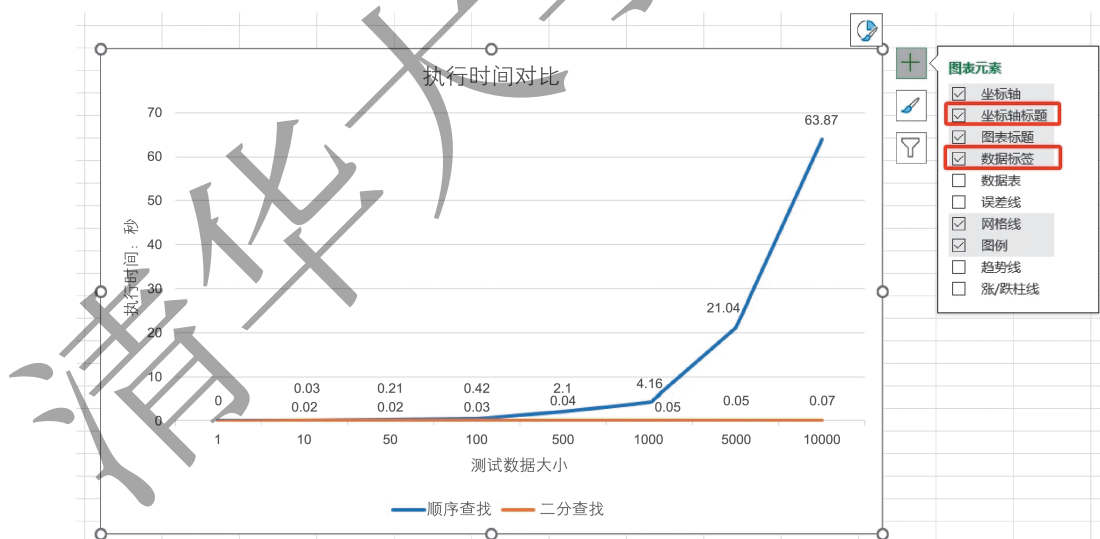


图 1.4.7 执行时间对比

小提示：

因为两种算法的数据差距巨大，从图表中很难看出数据的值，这时候



可以选中图标，单击右侧的[+]标签，在出现的选项卡中勾选“数据标签”复选框，让图形上显示准确的数据，勾选“坐标轴标题”复选框，并修改标题文字，让图表更具可读性，如图 1.4.7 所示。

5. 形成算法效率分析报告（表 1.4.3），分享交流

表 1.4.3 算法效率分析报告

比较项	顺序查找	二分查找
算法原理		
数据要求		
优点		
缺点		
复杂程度		
是否适合动态数据		
适用场景		



评一评

同学们，在本课中我们学习了如何对比算法的效率，请根据表 1.4.4，对学习过程进行评价，并在反思中不断成长与进步吧！

表 1.4.4 评价表

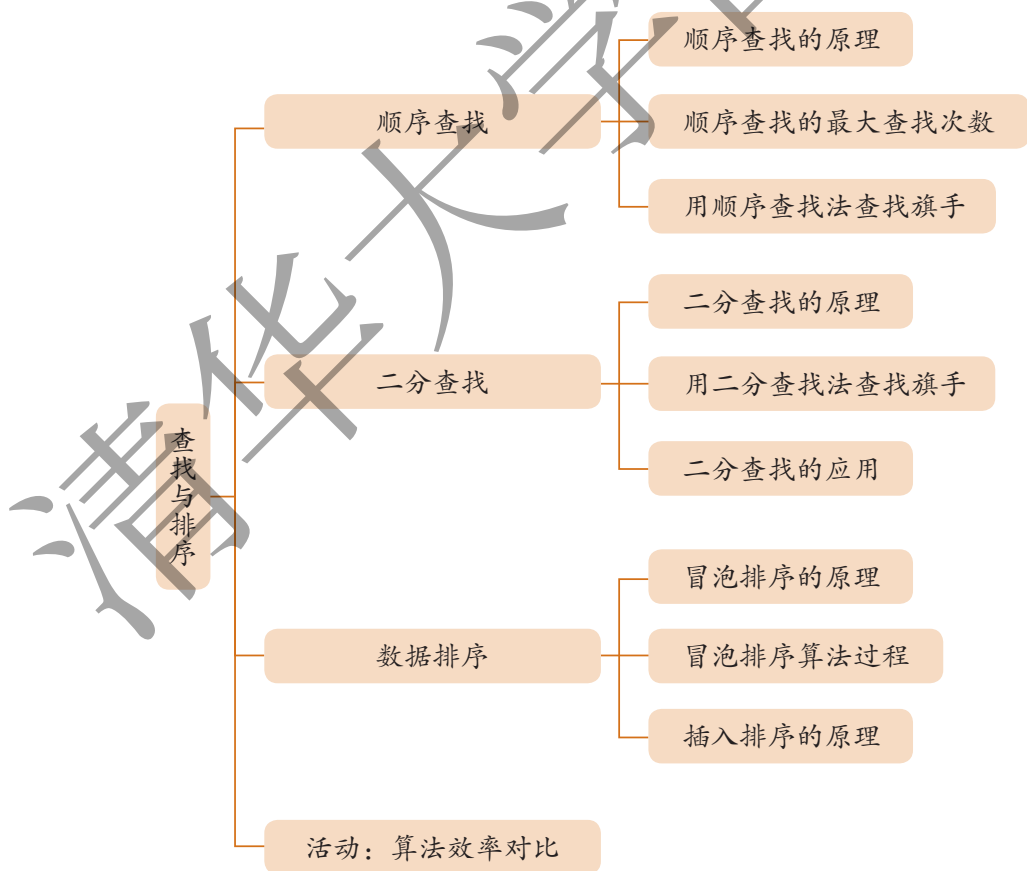
内容	遇到的问题	解决方法	感悟收获	总体评价
读一读				☆☆☆☆☆
议一议				☆☆☆☆☆
做一做				☆☆☆☆☆

小结与评价

本单元你学习了：

- ◆ 顺序查找
- ◆ 二分查找
- ◆ 冒泡排序
- ◆ 插入排序
- ◆ 算法效率对比

知识梳理





学习检测

1. 顺序查找与二分查找有什么区别?
2. 列举几个生活中应用二分查找思想的例子。
3. 冒泡排序与插入排序有何不同?
4. 一般从哪些方面对比算法效率?



反思评价

在本单元学习过程中,肯定少不了与他人互动交流、参与作品制作等活动。现在请就此进行总结与反思,以便更好地促进自身成长。

1. 从同伴那里学到了什么?
2. 向同伴分享了哪些观点?
3. 所运用的工具和方法是否得当?
4. 学习的过程中是否有值得推广的经验?

2

经典算法设计

📖 本单元你将学习：

- 枚举算法思想及其算法设计
- 密码的安全性能与其决定因素
- 递推算法思想及其算法设计
- 递归算法思想及其算法设计
- 函数调用自身的算法逻辑及程序验证
- 斐波那契数列的算法逻辑及其程序验证

在生活中，算法思想能让我们从很多相似的问题中找到一个通用的解决办法，就像用一个“万能钥匙”解决同一类问题。在设计算法的过程中，我们可以学会更清楚地思考和推理问题。

本单元，我们将继续使用算法思想解决生活中更复杂的问题，这些都是计算机科学中很经典的处理信息的案例，需要我们更有创造力地想出新点子。在此过程中，我们还会用到数学和编程的知识。我们可能会遇到困难，此时需要我们多次尝试和调整，希望同学们能有耐心、不放弃地继续挑战！



学习热身

同学们，请讨论一下，下面这些解决问题的方法都是算法吗？



家里买了一大筐苹果，我想知道哪个是最好吃的，于是我把每个苹果都咬了一口，一个一个比较，最后找到了我认为最好吃的那一个。

枚举算法 递推算法
递归算法
斐波那契数列
黄金比例

我打算从这个学期开始，第一天存1元零花钱，之后每天都比前一天多存1元钱，坚持一个月，我一共能存多少钱呢？应该怎么计算呢？



爸爸给我买了一个俄罗斯套娃玩具，我不知道它一共套了多少个，只能一个一个地打开，看看里面还有没有其他的，直到全部打开为止。知道了总数之后，还要一个一个地按顺序合上，直到恢复原貌。



你知道吗？

经典算法种类繁多，除了经典的二分查找、排序算法外，还有递推算法、递归算法、分治算法、贪心策略等，这些算法不仅可以帮助我们提高计算效率，解决实际问题，还能促进科学研究和技术创新，在各个领域都有着广泛的应用。

第1课

枚举算法



本课中你将学习：

- 什么是枚举算法
- 用枚举算法解决问题
- 密码的安全性与密码复杂度的关系

枚举算法是一种简单但功能强大的算法。在第一单元中，用顺序查找的方式找到符合身高要求的学生应用的也是枚举算法的思想。

尽管枚举算法在解决复杂问题时可能效率不高，但它对于一些小规模问题或解集有限的问题是非常有效的，同时也是理解和构建更复杂算法的基础。虽然有时候这样会很慢，但是只要有耐心，总能找到我们想要的结果。

一、逐个尝试的枚举算法

枚举算法也被称为穷举法或完全归纳法，是一种简单直接的解题方法。它的基本思想是通过一一列举问题所涉及的所有可能情况，然后对这些情况进行逐一分析，从而找出问题的所有可能解或证明问题无解。这种方法适用于那些可能情况数目有限且易于一一列出的问题。

枚举法的解题步骤如下。

(1) 明确问题：首先，需要清楚地理解问题的要求和所给条件，确定需要枚举的对象和范围。



(2) 逐一列举：根据问题的要求，不遗漏、不重复地列出所有可能的情况或解。

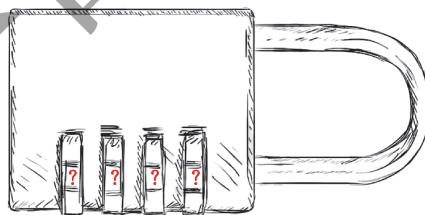
(3) 分析判断：对每一种列举出的情况进行仔细的分析和判断，看它是否符合问题的要求或条件。

(4) 得出结论：根据分析判断的结果，得出所有可能的解或证明问题无解。



一个假期过去了，我忘记了教室里个人储物柜密码锁的密码，你能帮我想办法解开密码吗？

这是一个长度为 4 位的纯数字密码锁，每一位都由 0~9 中的一个数字组成，各位数字之间没有规律。



想一想

1. 符合这种组成规律的密码最多有多少种？
2. 我们可以用随机猜想密码的方式来破解密码锁吗？
3. 如果要借助计算机帮忙，我们需要计算机怎么做？

这种密码锁的每一位都有 0~9 共 10 种可能的数字。第一位有 10 种可能，第二位又有 10 种可能，前两位可能的数字组合总数是 10×10 。同理，第三位又有 10 种可能，则前三位共有 $10 \times 10 \times 10$ 种数字组合，所以四位的数字密码锁，共有 $10 \times 10 \times 10 \times 10$ 种可能的密码组合。推导过程如图 2.1.1 所示。

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

只有一位的密码锁共有 10 种可能

第一位	第二位
0	0, 1, ..., 9
1	0, 1, ..., 9
2	0, 1, ..., 9
...	...
9	0, 1, ..., 9

有两位的密码锁共有 10×10 种可能

第一位	第二位	第三位
0	0, 1, ..., 9	每一个确定的第二位，都对应着 10 种第三位
1	0, 1, ..., 9	
2	0, 1, ..., 9	
...	...	
9	0, 1, ..., 9	

有三位密码锁的密码共有 $10 \times 10 \times 10$ 种可能

第一位	第二位	第三位	第四位
0	0, 1, ..., 9	每一个确定的第二位，都对应着 10 种第三位	每一个确定的第三位，都对应着 10 种第四位
1	0, 1, ..., 9		
2	0, 1, ..., 9		
...	...		
9	0, 1, ..., 9		

有四位密码锁的密码共有 $10 \times 10 \times 10 \times 10$ 种可能

图 2.1.1 四位密码锁共有多少种可能性

下面，我们设计一个随机生成 4 位数字密码的算法，并用程序进行验证。



(1) 随机数生成语句。

利用图形化编程软件，在“运算”模块中找到在一定范围内生成随机数的语句，如图 2.1.2 所示。



图 2.1.2 生成随机数语句

(2) 设置随机数生成范围。

我们要求生成的密码每一位都是从 0~9 中随机选取一个整数，因此将随机语句的取值范围设置成“在 0 到 9 间随机整数”，如图 2.1.3 所示。



图 2.1.3 在 0 到 9 间随机整数

(3) 输出 4 位纯数字密码。

利用“对话”语句，并排输出由 4 位 0~9 的随机整数组成的密码，如图 2.1.4 所示。



图 2.1.4 随机生成 4 位纯数字密码

二、用枚举法破解密码

利用枚举法——尝试、逐个排除的思想，我们可以设计出使计算机执

行时不重复、不遗漏的解密方式，帮助破译密码。我们已经知道 4 位数字密码一共有 10000 种组合方式，即 0000~9999，这里为方便计算，简化成数字 0~9999，缺失的数位可以用 0 补全。我们设计的算法可以从 0 开始尝试，然后是 1, 2, 3, ..., 9997, 9998, 9999，直到试出正确的密码为止。

探索

请你填写图 2.1.5 中缺少的内容，设计 4 位纯数字密码的破译流程图。

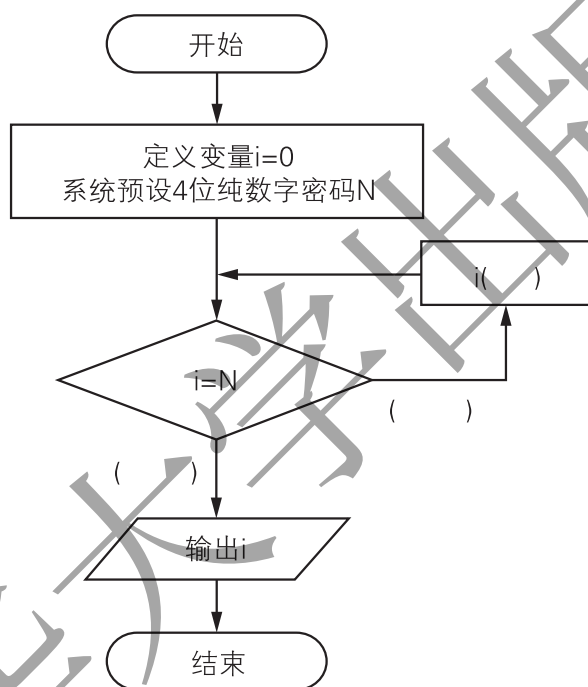


图 2.1.5 枚举法破译 4 位数字密码流程图

阅读

暴力破解 (brute force attack) 是一种通过枚举所有可能的密码或密钥组合，逐一尝试来破解密码或获取访问权限的攻击方法。它不依赖于特定的漏洞或技巧，而是纯粹利用计算能力进行大量尝试，直到找到正确答案为止。暴力破解方法虽然简单，但因需要尝试每一种可能性，破解过程通常十分耗时，尤其当密码较长或较为复杂时。现代的安全系统通常会通过限制尝试次数或增加密码复杂度来抵御暴力破解攻击。



三、从枚举算法看密码安全

枚举算法的优点在于它能够保证找到问题的所有解（如果存在的话），而且思路简单明了，易于理解 and 操作。然而，枚举算法也有其局限性，特别是当问题的可能情况非常多时，枚举所有情况可能会非常耗时和费力，甚至在实际操作中变得不可行。

因此，在使用枚举法时，需要权衡问题的规模和复杂度，以及枚举所需的时间和资源。在某些情况下，可能需要结合其他更高效的算法或方法来优化解题过程。

当编码的规律被掌握之后，用枚举算法就一定能尝试找出正确的解码答案，所以在密码学上，编码规则被视为最高机密，并经常更新，以防止时间久了以后，密码被破译。

阅读

恩尼格玛机（图 2.1.6）是一种电动机械转轮密码机，在“二战”初期提供了几乎无法破译的通信保障，使得德国在情报战中占据绝对优势。面对恩尼格玛机这一看似无懈可击的加密设备，艾伦·麦席森·图灵（图 2.1.7）展现出了非凡的智慧和勇气。他设计并建造了名为炸弹机（Bombe）（图 2.1.8）的机器，利用恩尼格玛机的加密原理进行反向操作，在进入穷举法前先排除一些自相矛盾的组合方式，然后将多台这样的机器环形相连，更快速地测试可能的密码设置，以找到正确的答案。

图灵和他的团队在密码战中的胜利不仅推动了密码学和计算机科学的发展，还提高了人们在信息安全方面的重视程度和防范意识。图灵因此被誉为计算机科学之父和人工智能之父。

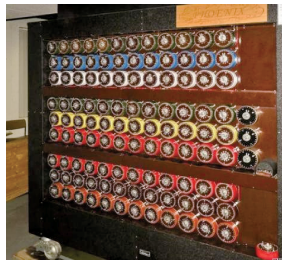
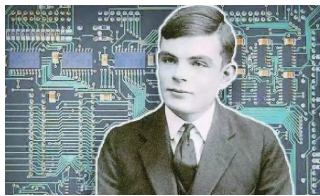
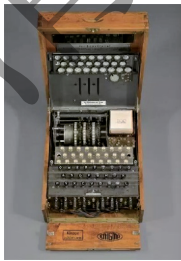


图 2.1.6 恩尼格玛机 图 2.1.7 艾伦·麦席森·图灵 图 2.1.8 炸弹机

探索

根据枚举法的工作原理，结合自己的在线生活经历，说一说在不增加密码位数的前提下，如果要提高密码的安全性，可以有什么方法？



想一想

4 位纯数字密码有 $10 \times 10 \times 10 \times 10$ 种组合，每增加一位密码组合增加 _____ 倍。

四位数字加小写字母组合的密码有 _____ 种组合，每增加一位密码组合增加 _____ 倍。

4 位数字加大小写字母加 @# ¥% 特殊符号组合的密码有 _____ 种组合，每增加一位密码组合增加 _____ 倍。

挑战

利用图形化编程软件，验证破译 4 位纯数字密码的流程图，挑战用枚举算法，实现不重复、不遗漏地进行密码破解。

第2课

递推算法



本课中你将学习：

- 什么是递推算法
- 用递推算法解决问题
- 根据目标制订合理的储蓄计划

递推算法是一种既巧妙又实用的解题方法，它就像搭积木一样，虽然每块积木都很小，但当我们按照规律，一块一块地搭上去，最后就能搭建出一座壮观的城堡。利用递推算法解决复杂问题，就是从最简单基础开始，按照规律逐步推算，一步步走向复杂，最终解决问题。



密码锁很方便，但还是有可能被暴力破解，我家的门锁也是密码的，这要是被破解了可就危险了。据说人的指纹几乎是独一无二的，我想把我家的门锁换成指纹锁。

我想买的指纹锁要360元，我应该怎么制订储蓄计划，才能完成这个购买目标呢？



360 元

一、有规律的递推算法

制订储蓄计划，用得最多的就是递推算法。递推算法是一种数学和计算机科学中常用的解题方法。通过已知的简单情况，也就是初始条件，以及一个或多个固定的规律，我们称为“递推关系式”，基于这种固定的关系逐步累积下去，就会得到更复杂的结果。简单来说，递推算法就是从已知的基础情况出发，按照某种确定的规则，一步一步地计算出问题的解。我们经常玩的数字接龙游戏，就是蕴含着递推算法。

数字接龙游戏的规则是这样的：第一个人说一个数字，比如 1；然后第二个人要根据第一个人的数字，加上一个固定的数（比如 2），然后说出结果，也就是 3；接着，第三个人再根据第二个人的结果，继续加上那个固定的数 2，说出 5……就这样，每个人都说出一个数字，而这个数字是前一个人的数字加上一个固定的数（图 2.2.1）。

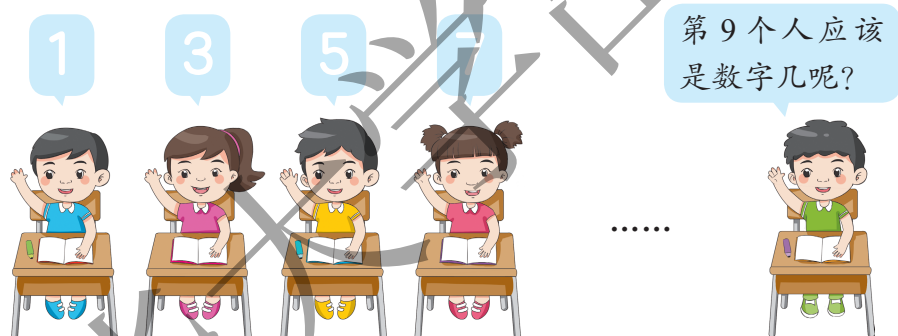


图 2.2.1 数字接龙游戏

数字接龙游戏其实就是一个简单的递推算法。在这个例子里，“递推”就是根据前面的结果和固定的规律，一步步推算出后面的结果。只要确定好初始条件和递推规则，就可以一直玩下去啦！

阅读

递推算法是一种用若干步可重复运算来描述复杂问题的方法，是序列计算常用的算法，通常需要根据前面的一些项通过计算得出序列中的指定项的值。其思想是把一个复杂的、庞大的计算过程转化为简单过程的多次重复，充分利用了计算机速度快和不知疲倦的特点。



探索

你能找出 1、2、4、8、…这列数的递推规则吗？

根据上面的递推规则，我们一起来玩一个小游戏（图 2.2.2）。

参与游戏的小组内同学，按照上面的递推规则依次（一人一次一个数）循环往复地报出符合递推规则的序列中的每一个数。

其他的同学作为裁判，当游戏中的小组报数时间间隔时间大于 3 秒或者报错数字的时候叫停。

换一个小组再次进行游戏。

最后报数最大的小组获胜。

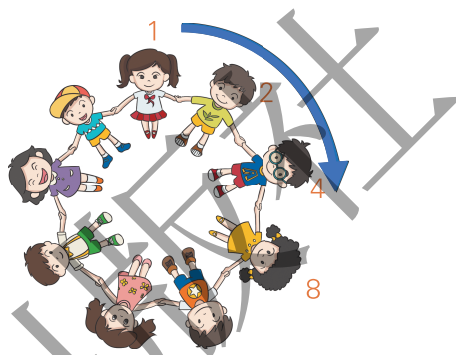


图 2.2.2 按递推规则报数

二、用递推算法制订储蓄计划

我们利用递推算法来推算一下小清的储蓄计划。



每个月存 10 元零花钱，我需要存 36 个月才能存够 360 元买下一个指纹锁呢。

爸爸支持你的计划！你每月存入 10 元，爸爸每月在你上月存款总额的基础上为你多存 20 元。



探索

根据小青爸爸的储蓄方案，小青每月是如何存钱的呢？小青每月存入 10 元，爸爸在小青上月存钱的基础上，每月再赞助小青 20 元，列表分析如表 2.2.1 所示。

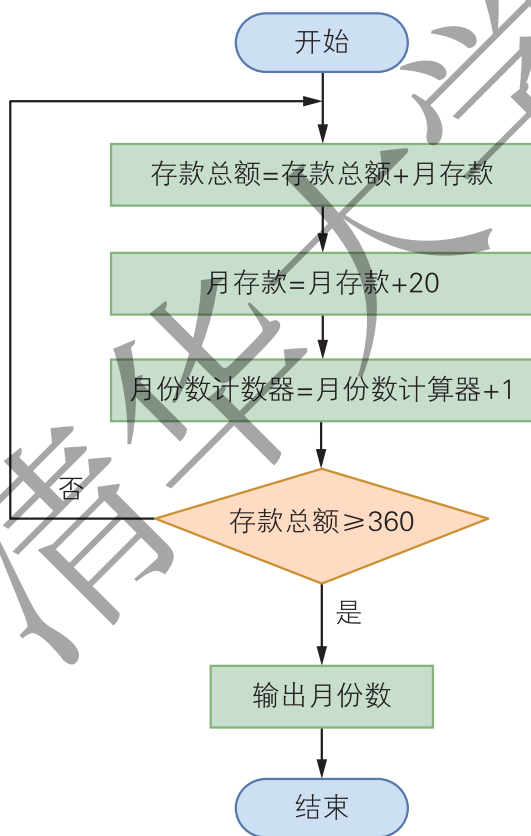
表 2.2.1 小青爸爸的储蓄方案

时间	第 1 个月	第 2 个月	第 3 个月	...
小青	10	10	10	...
爸爸	0	10+20	40+20	...
每月储蓄合计	10	40	70	...

每月储蓄合计金额分别为：10、40、70...

小青爸爸储蓄方案中每月储蓄合计金额的递推规则是：_____。

按照这个方案，存够 360 元需要多长时间？我们用流程图（图 2.2.3）分析如下。



想一想

在图 2.2.2 中，存款总额 = 存款总额 + 月存款 的含义是_____，月存款 = 月存款 + 20 的含义是_____，月份数计数器 = 月份数计数器 + 1 的含义是_____。

流程图中，_____情况下，输出月份数，程序结束；_____情况下，程序循环计算每月储蓄。

图 2.2.3 递推算法计算储蓄月份流程图



流程图我看懂啦，但是究竟要用多少月才能存够 360 元呢？

三、编程实现递推算法

用图形化编程软件实现递推算法的程序中，先新建三个变量分别是月存款、存款总额、月份数计数器，再将变量初始化，接下来该怎么进行数据的迭代、判断呢？

也可以用“重复执行直到 ()”语句，判断循环终止的条件是：存款总额 = 360。

可以使用重复执行，再判断存款总额是否大于 360。

在重复执行中一定要把月份数计数器增加 1。

我觉得判断是否存够的时候用大于和等于更严密。



综合小组内同学们的意见，首先新建三个变量，然后将变量初始化，如图 2.2.4 所示。



为什么月存款的初始值是 10 呢?

图 2.2.4 变量初始化

接下来, 就使用“重复执行直到 ()”的积木来实现数据的迭代, 如图 2.2.5 所示。



将变量增加的含义是什么?

图 2.2.5 数据的迭代

最后, 可以用对话来把存款达到 360 元所需的月数显示出来。

探索

用图形化编程软件搭建程序, 算出小青按照爸爸给的储蓄方案存够 360 元需要多少个月?

挑战

课后了解银行现行的各类储蓄的利率, 以 10000 元为本金, 利用递推算法设计程序, 计算一年之后不同的储蓄方法会得到多少利息, 找到获益最高的储蓄方法, 给爸爸妈妈提供家庭储蓄建议。

第3课

递归算法



本课中你将学习：

- 什么是递归算法
- 设计递归函数
- 递归解决棋盘格放麦粒问题

存够了购买指纹锁的钱，小青带着存钱罐早早地来到专卖店购买心仪的指纹锁。他发现商家贴出了购买指纹锁限量赠送 100 套电子猫眼的活动，在活动现场排了好多人，小青也加入了排队大军中（图 2.3.1）。小青想知道目前所排的位置能否获得赠送电子猫眼的资格。



图 2.3.1 排队购买指纹锁

大家都在排队，也没人有空去清点人数。小青想知道前面有多少人在排队，你有什么好办法吗？

一、环环相扣的递归算法

小青不能离开队伍，要知道前面有多少人，只能问前面的人，而他前面的人说：“我也不知道前面排了多少人，怎么办呢？”小青开动脑筋想了一个办法：请大家传话下去“你是第几个？”直到传到第一人，然后再依次返回“我 = 我前面的人数 + 1”的消息，这样大家都知道前面有多少人在排队了。（图 2.3.2）



图 2.3.2 图说“递归”

探索

小青的办法可行吗？我们也来试一试。

排队的人们，都在问同一个问题：_____。

传递消息的终止条件是：_____。

返回消息的时候，人数计算方法是：_____。

小青的方法中，大家一直在重复做一件事：不断地询问前面的人，直到问至第一人，询问停止，然后将前面的人数加上自己再传给后面的人。

把一个大问题分解成更小的问题，并且这些更小的问题与原问题相似。它会一层层缩小问题的规模（递归调用），直到遇到最小、最简单的情况，这个时候可以直接得到答案（也称为“基准情况”或“终止条件”）。这种解决问题的思维在计算机科学中叫作递归算法思想。



二、递归算法的实现——递归函数

递归函数是指在定义中调用自身的函数。递归函数用来解决可以分解为相似小问题的任务，通过逐步缩小问题规模，直到达到一个简单的“基准情况”或“终止条件”，停止调用自身并返回结果。

阅读

在编程中，函数是一段独立的代码，用来执行特定的任务或计算。可以把函数想象成一个“小机器”或“工具箱”，我们可以给它提供一些输入（称为参数），它会根据设定好的步骤进行处理，最后返回一个结果（称为返回值）。

函数有以下几个关键特点。

（1）封装性：函数将一段代码封装起来，方便重复使用，避免重复写同样的代码。

（2）输入参数：有些函数需要输入参数。不同的输入会让函数产生不同的输出。

（3）输出结果：函数执行完任务后，通常会产生一个结果，这个结果可以传递给程序的其他部分。

使用函数的好处是让代码更简洁，组织更清晰，便于理解和维护。

图 2.3.3 中定义了一个名为“hello”的函数，并且有一个名为“userName”的输入参数，函数中将“你好”与“userName”放在一起返回。

当“开始被点击”时，调用函数“hello”，输入参数“小清”，通过对话显示出返回值为：你好小清。



图 2.3.3 函数的定义与调用

我们可以用自然语言将上面解决问题的方法写为一个递归函数。

函数 **统计人数函数**(n): 函数名为“统计人数函数”，有一个传入参数 n

如果 $n=1$: $n=1$ 时，就是消息传到了队伍第一人，函数返回 1 并终止执行

返回 1

否则: 否则，返回调用自身“统计人数函数”，不同的是，传入参数减少了 1，并且将结果加 1

返回 **统计人数函数**($n-1$)+1

我们可以用图形化编程软件编写上面的过程（图 2.3.4）。

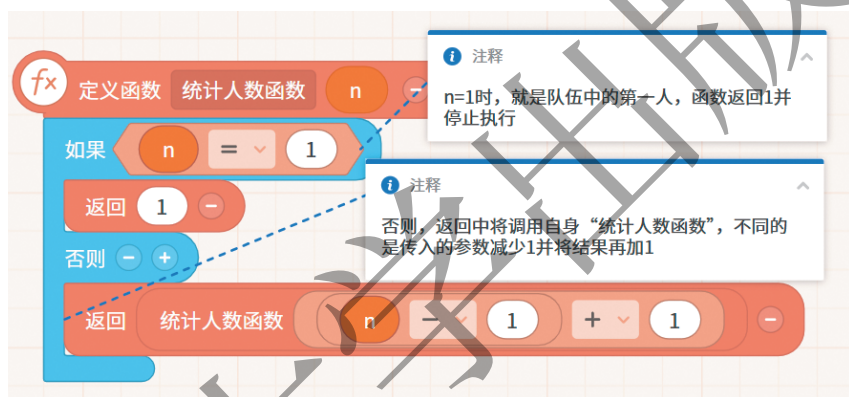


图 2.3.4 递归函数

探索

试着代入函数的参数，比如 $n=3$ ，说一说函数是怎么运行的？调用自身的时候 n 为什么要减 1，返回的结果又为什么要加 1？

三、经典的递归算法——棋盘里的麦粒

印度有一个古老的传说：舍罕王打算奖赏国际象棋的发明人——宰相西萨·班·达依尔。国王问他想要什么，他对国王说：“陛下，请您在这张棋盘的第 1 个小格里，赏给我 1 粒麦子，在第 2 个小格里放 2 粒，第 3 小格放 4 粒，以后每一小格中麦粒的数量都比前一小格加一倍。请您把这样摆满棋盘上所有的 64 格的麦粒，都赏给您的仆人吧！”国王觉得这个要求太容易满足了，就答应给他这些麦粒。当人们把一袋一袋的麦子搬来开始



计数时，国王才发现：就是把全印度甚至全世界的麦粒全拿来，也满足不了那位宰相的要求。那么，宰相要求得到的麦粒到底有多少呢？

这个问题可以用很多方法来解答，之前学习的递推算法以及本课中的递归算法都可以解决这个问题。递归无疑是最简洁的方法，我们一起来设计递归算法，解决棋盘里的麦粒问题吧。

西萨宰相要求国王往棋盘格里放麦粒的规则是：每一小格都比前一小格加一倍，如表 2.3.1 所示。

表 2.3.1 麦粒摆放规则

棋盘格	1	2	3	...	64
麦粒	1	2	4	...	
规律	1	前一格 $\times 2$	前一格 $\times 2$		

探索

你能填写图 2.3.5 中的图形化编程递归函数调用自身算法中黑色圆形内的数字与计算符号吗？

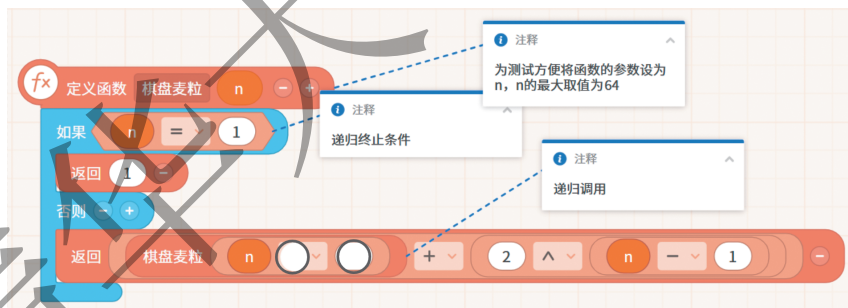


图 2.3.5 递归计算棋盘格麦粒问题

挑战

用图形化编程软件搭建递归计算棋盘格麦粒问题的函数，并代入参数 64，计算出麦粒的数量。如果一颗麦粒的重量是 0.03 克，国王一共要给宰相多少千克的麦粒呢？

第4课

跨学科活动：斐波那契数列与黄金比例



本课中你将体验：

- 大自然神奇的图案中蕴藏的秘密
- 理解斐波那契数列的基本概念
- 用算法计算斐波那契数列
- 斐波那契数列与黄金分割比例

你有没有想过，为什么向日葵的种子排列得那么完美？为什么蜗牛的壳总是旋转得那么优雅？其实，大自然里藏着许多秘密，而其中一个非常特别的秘密就藏在一串神奇的数字里——斐波那契数列（图 2.4.1）！

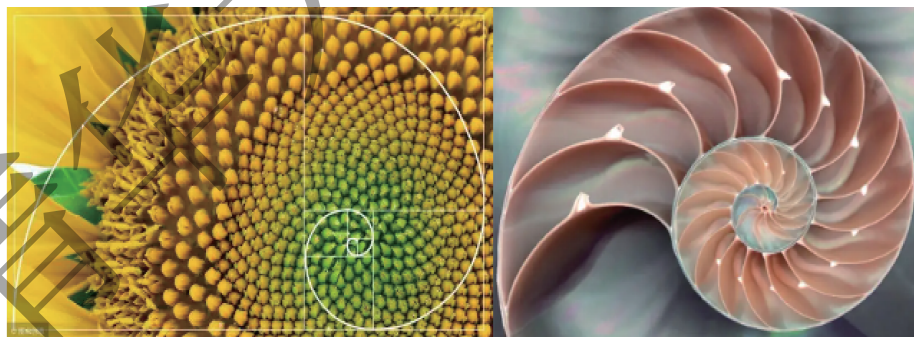


图 2.4.1 优雅的造型中隐藏的斐波那契数列

这不是一串普通的数字，而是一串有着特别规律的数字序列，它们能帮助我们解释自然界中的很多现象。今天，我们就一起来探索这个神奇的数列，看看它如何从数字的世界走进现实，给我们带来如此多的惊喜！

探索斐波那契数列，我们从著名的“兔子问题”开始吧。



想一想

意大利数学家斐波那契 (L.Fibonacci) 在他 1202 年出版的《算盘书》一书中记述了有趣的兔子问题：假定一对刚出生的小兔子从出生后的第 3 个月起每个月都生一对小兔子（假定都是一公一母）。第 1 个月有一对刚出生的小兔子，12 个月后有多少对兔子？

用列表表示如表 2.4.1 所示。

表 2.4.1 每个月兔子的对数

月数	1	2	3	4	5	6	7	8	9	10	11	12
兔子对数	1	1	2	3	5	8	13	21	34	55	...	?

- 第 1 个月：只有 1 对兔子（刚开始的那对）。
- 第 2 个月：还是 1 对兔子。
- 第 3 个月：有了 2 对兔子（最初的那对生了一对小兔子）。
- 第 4 个月：有了 3 对兔子（最初的那对又生了一对）。

.....

这个数字越来越大，你能看出来其中的奥秘吗？



算一算

计算表 2.4.2 中的算式，想一想斐波那契数列有什么规律。

表 2.4.2 斐波那契数列中的规律

月数	1	2	3	4	5	6	7	8	9	10
兔子数	1	1	2	3	5	8	13	21	34	55
将前两月的兔子数相加的值			1+1 =	1+2 =	2+3 =	3+5 =	5+8 =	8+13 =	13+21 =	21+34 =
前值除以后值						5 ÷ 8 =	8 ÷ 13 =	13 ÷ 21 =	21 ÷ 34 =	34 ÷ 55 =



- (1) 斐波那契数列中，前面两个数之和等于_____。
- (2) 斐波那契数列中越靠后的数除以它后面的数，商越接近_____。
- (3) 斐波那契数列中前4个数之和比第6个数_____。



做一做

斐波那契数列中，每个数字都是前两个数字之和，根据这一规律，选用递归算法来求出斐波那契数列的值是最简洁的方法。从表 2.4.2 得出，此递归函数的终止条件是传入的参数值 n 小于或等于 2 时则返回 1。

1. 用自然语言描述递归算法代码

函数 斐波那契 (n)

当 n 小等于 2 时

返回 1

否则：

返回 斐波那契 ($n-1$) + 斐波那契 ($n-2$)

2. 用图形化编程软件搭建斐波那契递归函数

(1) 定义函数。

在图形化编程软件中单击 ，选择定义函数积木，将函数名称修改为“斐波那契”，然后单击函数后面的+号，为函数增加一个参数 n ，如图 2.4.2 所示。



图 2.4.2 定义斐波那契函数

(2) 搭建完整的递归代码。

搭建递归函数的时候，一定注意递归的基准情况与递归条件，二者缺一不可，如图 2.4.3 所示。调用函数的时候， n 值不宜取值过大，否则可能会由于软件的限制无法运行。

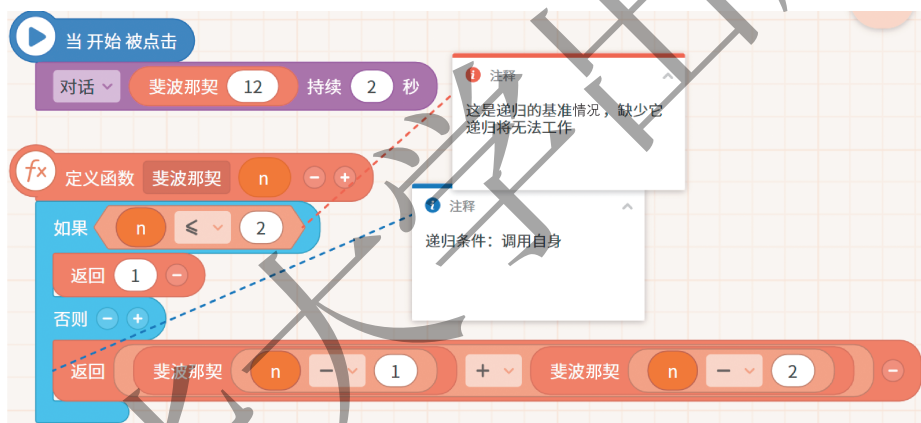


图 2.4.3 图形化编程搭建递归计算斐波那契数列

将 n 的不同取值运算结果填在表 2.4.3 中。

表 2.4.3 运算结果

n 值	1	2	3	4	5	6	7	8	9	10	11	12
数列值												

(3) 探究斐波那契数列前值除以后值。

将调用代码稍加修改，让斐波那契数列的前值除以后值，如图 2.4.4 所示。



图 2.4.4 斐波那契数列前值除以后值

将实验结果（保留小数点后 3 位）填写在表 2.4.4 中。

表 2.4.4 实验结果

$(n=7) \div (n=8)$	$(n=8) \div (n=9)$	$(n=11) \div (n=12)$	$(n=14) \div (n=15)$	$(n=20) \div (n=21)$

我们会惊奇地发现，实验结果是无限不循环小数，近似值为_____。



读一读

0.618 是一个特别而优雅的数字，它也被称为黄金分割比。

黄金分割是指将整体一分为二，较大部分与整体部分的比值等于较小部分与较大部分的比值，其比值约为 0.618。这个比例被公认为最能引起美感的比例，因此被称为黄金分割比。

在图 2.4.5 中，达·芬奇的《蒙娜丽莎》中头肩比例，埃及胡夫金字塔的边长与塔高之比，都接近 0.618 这个黄金分割比。



图 2.4.5 《蒙娜丽莎》和胡夫金字塔中的黄金分割比



用斐波那契数列中的值作为正方形的边长，前面两个正方形的边长之和恰好是下一正方形的边长，两个相邻正方形的边长之比如图 2.4.6 中红色线段与蓝色线段的比例永远接近黄金分割比例 0.618。

以每个正方形的边长为半径画圆形成的螺旋被称为黄金螺旋，它们的关键点都落在了黄金分割点上，形成的构图也极具美感。

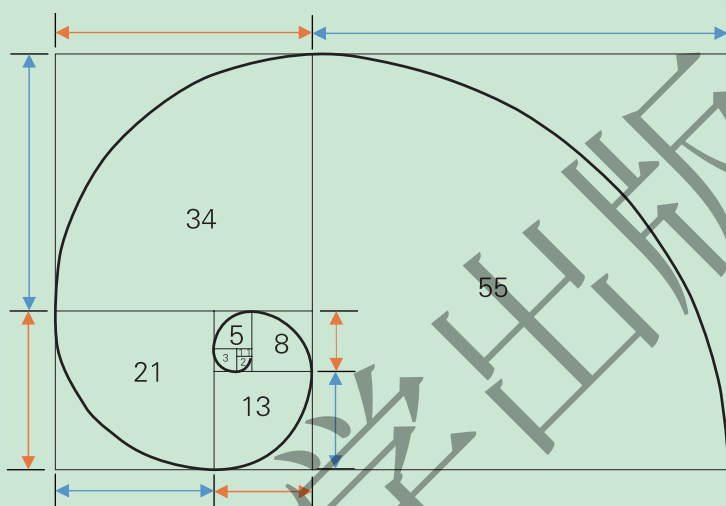


图 2.4.6 黄金螺旋

这也揭秘了本课开始的两张极具美感的图片中的奥秘。



评一评

同学们，在本课中我们学习了斐波那契数列，请根据表 2.4.5，对学习过程进行评价，并在反思中不断成长与进步吧！

表 2.4.5 评价表

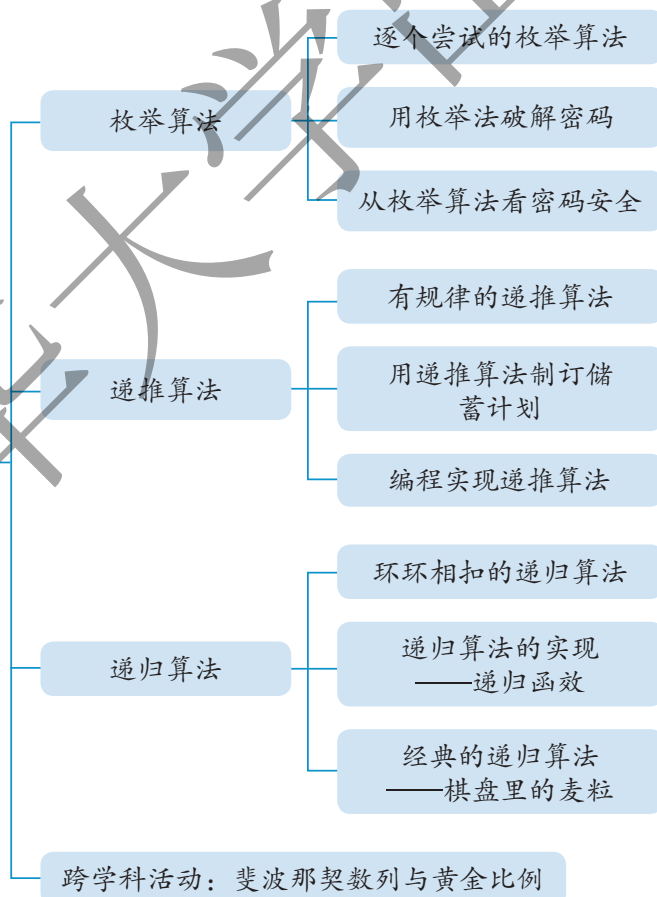
内容	遇到的问题	解决方法	感悟收获	总体评价
想一想				☆☆☆☆☆
算一算				☆☆☆☆☆
做一做				☆☆☆☆☆
读一读				☆☆☆☆☆

小结与评价

本单元你学习了：

- ◆ 枚举算法
- ◆ 递推算法
- ◆ 递归算法

知识梳理





学习检测

1. 结合暴力破解，谈谈如何增加密码的安全性？

2. 说一说递归算法为何要设一个明确的终止条件？



反思评价

在本单元学习过程中，肯定少不了与他人互动交流、参与作品制作等活动。现在请就此进行总结与反思，以便更好地促进自身成长。

1. 从同伴那里学到了什么？

2. 向同伴分享了哪些观点？

3. 通过本单元的学习，学到哪些解决问题的思路？

4. 设计的程序是否可以顺利执行？

5. 调试程序的过程中有哪些成功的经验可以分享给大家？

3

人工智能与算法

📖 本单元你将学习：

- 智能推荐算法的原理、优点及不足
- 智能导航算法
- 智能识别的原理及认知 AI 的使用
- 体验生成式人工智能

人工智能简称 AI，是一种让计算机和机器变得“聪明”的技术。就像人类有大脑可以思考、学习和解决问题，人工智能也能让计算机通过学习很多数据，来完成各种任务。比如，人工智能可以帮助手机识别你的脸、让机器人学会对话，还能帮医生更快地找到病因。在我们的生活中很多地方都有 AI 的帮助，它就像一个“聪明小助手”，让我们的生活更加方便、有趣！

本单元，我们将探索智能推荐、智能导航算法的秘密，体验强大的认知 AI 和生成式人工智能。



学习热身

同学们，人工智能已经悄悄融入了我们的学习、工作和生活。

- 每天进出小区或学校，门禁系统总能准确识别并为我们开门放行。

- 面部识别不仅能认出我们，还能识别我们的情绪。

- 不论道路多复杂，导航软件总能帮我们找到最快捷的路线。

- 我们遇到不会的题目，除了可以问老师，也可以向 AI 求助。

请你根据生活经验，说一说人工智能技术在生活中有哪些应用，并试着解释其中的原理。

智能推荐
智能导航
表情识别
生成式人工智能



你知道吗？

人工智能正在快速发展，未来的世界会越来越依赖人工智能。谁能掌握人工智能，谁就会在各个领域有更多的发展机会。但是同时我们也面临着一个残酷的问题：未来 20 年，近半数的人类工作会被智能机器所替代。

第1课

智能推荐



本课中你将学习：

- 智能推荐算法的基本原理
- 智能推荐的应用效果
- 智能推荐的优点及不足

当你打开一些视频软件或者购物网站的时候，是不是经常能看到软件或网站推荐的视频或者商品？这些推荐可不是随便来的哦，它是智能推荐系统根据你观看视频的习惯和平常购物的习惯给你推荐的。智能推荐系统就像是一个特别贴心又聪明的小助手，努力地猜着用户可能会喜欢什么。

一、智能推荐算法

在众多视频软件里，短视频软件堪称最易使人沉迷的一类，很多人一旦拿出手机打开短视频进行观看，短则几分钟，长则几小时的时间就从指缝中溜走了。那这些短视频软件究竟有什么样的魔力，可以让人如此“爱不释手”呢？

它会根据我在每个视频上停留的时间为我推荐视频。

它会根据我的点赞和评论记录为我推荐视频。

它会根据我的搜索或浏览记录为我推荐视频。



它会根据我在短视频平台上的消费记录为我推荐视频。



通俗点讲，智能推荐算法就是通过我们曾经发生的行为痕迹去推测我们当下和未来所需。我们以短视频软件为例，逐步分析和理解智能推荐算法的几个重要步骤。

1. 采集用户数据

用户使用短视频软件浏览视频的过程就是一个不断选择的过程。比如：你看完一个美食博主的探店视频并点了赞，而一个经济学普及的视频刚播放了不到 2 秒，你就划走了，在这个过程中，系统就完成了一次信息收集。数据类别如表 3.1.1 所示。

表 3.1.1 数据类别

数据类别	具体内容
内容类数据	时政、健康、艺术、美食、商品评测、游戏攻略等
用户类数据	性别、年龄、职业、所在的区域等
操作类数据	视频完播、点赞、评论、转发、收藏等

当然，短视频软件不仅会收集我们的观看使用习惯数据，也会同时收集平台上和我们年龄差不多的用户或者和我们有相同观看喜好的人的使用习惯数据。

2. 数据分析构建用户画像

收集完这些数据后，智能推荐算法就要开动脑筋找规律啦。它会使用一些很厉害的分析算法，常用分析算法如表 3.1.2 所示。

表 3.1.2 常用分析算

算法名称	具体内容
神经网络算法	根据用户历史喜欢的内容，为用户找出与他历史喜欢的内容相似或相关的内容
协同过滤算法	依赖于“群体共性”“群体智慧”，挖掘出那些潜在的、可能会被用户喜欢的内容
关联规则推荐	从本质上讲，它类似于协同过滤算法，只是它协同的是用户自己的购买记录

当系统完成数据分析后，会生成一个“用户画像”（图 3.1.1），用来描述用户特征的信息，模拟真正的用户行为。

3. 推荐适合的视频

经过数据收集、数据分析并形成用户画像，智能推荐算法就可以给我们做出推荐啦。它会把它觉得我们可能会喜欢的视频，放在一个很显眼的地方。



图 3.1.1 用户画像

探索

请根据你的个人信息和兴趣爱好，在图 3.1.2 的格子里填上信息，完成你的“用户画像”，方便大家了解你。

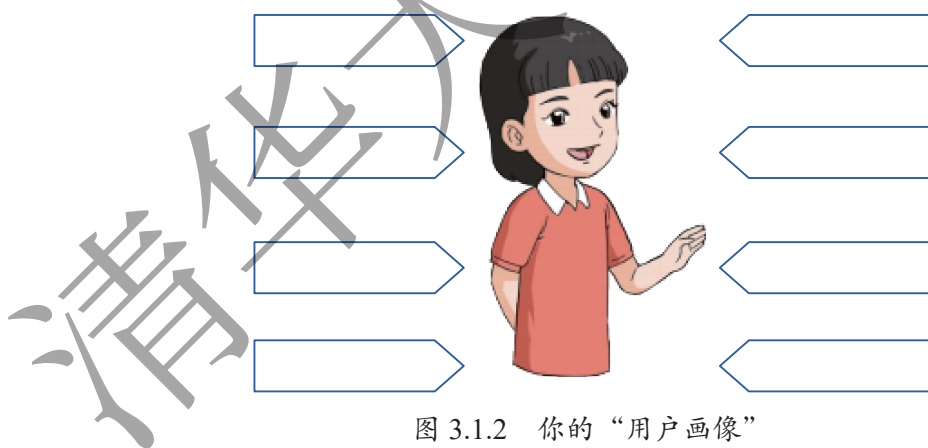


图 3.1.2 你的“用户画像”

根据我的“用户画像”，我认为平台会给我推送_____的视频内容。

智能推荐算法除了应用于短视频平台和购物平台外，还广泛应用于各

种不同的场景和领域。请你在表 3.1.3 中填写你还在哪些场景和领域体验过智能推荐算法。

表 3.1.3 智能推荐算法应用场景和领域

场景和领域	应用
短视频平台	根据用户的浏览习惯，推荐满足用户需求的视频内容

二、体验智能推荐

同学们，常用的输入法也有类似于智能推荐的功能。我们一起来体验一下吧！



我们常用的输入法也通过智能推荐算法让用户得到更好的体验。

当我们在使用输入法打字的时候，输入法常常会给我们推荐一些词语、句子或者表情符号等，这就是输入法中的智能推荐。它就像是输入法里的小助手，帮助我们在打字时打得更快、更方便。

以输入词语“遗址”为例。

第一次输入拼音：yi zhi，如图 3.1.3 所示，输入法给出的第一页备选字词中没有这个词语，我们可以通过翻页，找到“遗址”这个词语并输入。



图 3.1.3 第一次输入拼音

再次输入拼音：yi zhi，如图 3.1.4 所示，输入法给出的第一页的第一个词语就是我们所需的“遗址”二字。



图 3.1.4 第二次输入拼音

通过连续输入训练，输入法就能够快速地将原本不在第一页的词语移动到第一页的第一条，其他的词语则依次往后顺移。

接下来，我们尝试将“遗址”“一致”“一直”“已知”“一只”和“一支”在文档中反复多次输入，我们还可以发现，输入法会根据我们使用每一个词语的频率（次数），将第一页的所有词语按照如图 3.1.5 所示的顺序排列。



图 3.1.5 多次输入不同词语后的排序效果

探索

想出一个字或者词语，一定要选有其他汉字或者词语和它是同音的才可以哟。然后，请你用计算机上的输入法试试看，你想出来的这个字或者词是否能够通过你的多次输入，成功排到输入法备选词的第一个。最后，请将你尝试后的情况记录在表 3.1.4 中。

表 3.1.4 字词输入记录表

字或词	你输入了几次后排到了第一个	是否成功

三、智能推荐并不完美

通过智能推荐，人们能更有效地获取到更为精准的信息，对于个人用户而言，它具有精准推荐和节省时间的优点，如表 3.1.5 所示。



表 3.1.5 智能推荐的优点

优点	具体内容
精准推荐	对用户的历史行为、兴趣偏好等数据进行分析，能够准确地知道用户的需求和兴趣
节省时间	可以帮助用户快速筛选出符合需求的信息，节约筛选和搜索所花费的时间，提高信息获取的效率

智能推荐技术确实给我们的生活带来了便利，但我们也需要关注它的不足之处，如表 3.1.6 所示。

表 3.1.6 智能推荐的不足

缺点	具体内容
信息茧房效应	智能推荐会让用户一直看到相似的内容，而难以接触到新的信息或观点
隐私安全风险	智能推荐系统会收集用户的浏览记录、搜索内容、地理位置等信息。这些数据一旦被不当使用或泄露，可能引发隐私问题和安全问题
缺少人性化	智能推荐能从行为数据中识别用户偏好，但不能理解用户的实际心理感受或情境需求
不能动态变化	智能推荐系统主要依赖用户的历史行为数据，如果用户兴趣在近期发生变化，但系统继续推荐原兴趣相关的内容，就可能出现推荐“不合时宜”的情况

智能推荐技术的发展仍然在不断优化，以平衡个性化和多样性、增强数据安全和隐私保护，并努力在偏见、个性化理解和动态变化的适应性上做出改进。这些改进将使智能推荐在未来更全面地服务用户的多样化需求。

挑战

现在大部分软件都允许用户关闭个性化推荐内容，请上网查询关闭的方法，尝试找到并关闭手机上部分软件的个性化推荐功能。

第2课

智能路径规划



本课中你将学习：

- 了解生活中的“最短路径”
- 用贪心策略解决最短路径问题
- 用狄克斯特拉算法寻找最短路径

大家在生活中应该都有过这样的经历吧：当我们要去一个不太熟悉的地方时，不知道怎样前往目的地。这时我们会怎么做呢？很多人都会拿出手机，打开导航软件，输入目的地，接着导航就会为我们规划出几条路线，告诉我们该怎么去。

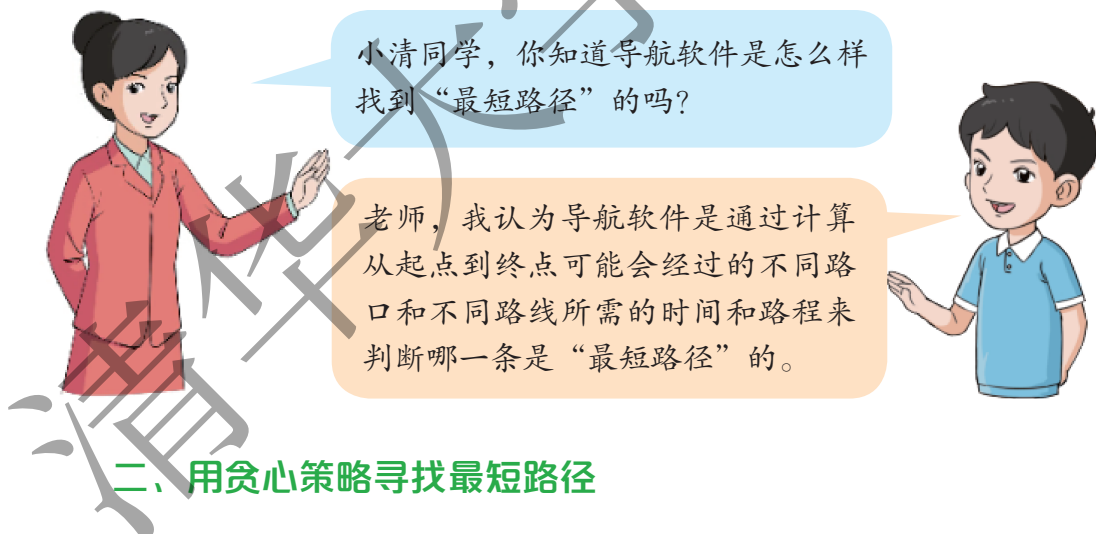
一、生活中的“最短路径”

以从乌鲁木齐站到乌鲁木齐地窝堡国际机场为例，导航软件给我们提供了三种驾车方案，有的方案看起来路线比较直，有的可能会稍微绕一点儿路（图3.2.1）。我们一般会选择哪条路线呢？

没错，我们通常会选择那条看起来能让我们最快到达的路线，也就是路上花费时间最少（推荐）或者路程相对较短（方案3）的路线。这其实就是我们生活中经常遇到的“最短路径”问题。



图 3.2.1 导航方案



导航软件找到最短路径的过程包含一些很有意思的算法和计算过程。其中有一种简单的算法策略叫作贪心策略：每次总是做出在当前看来是最好的选择，希望通过一系列的当前最优选择，最终得到一个整体上比较满意或者接近最优的结果。

给你一张如图 3.2.2 所示的有向图（只能按照箭头所指的方向行进，不可逆行），假设你想从起点 A 去往 D 点，每一个圆圈代表途经的一个节点（路口），每一个带箭头的线段代表一段路，线段上的数字表示这段路的长度，单位为千米。请你观察路线图并写出最短路径。

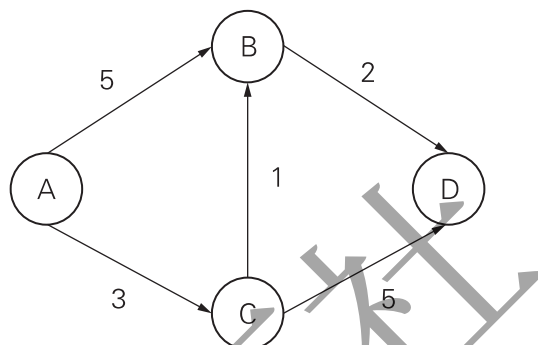


图 3.2.2 路线图

A→

在这个路线图中，我们使用贪心策略成功找到了从 A 点到 D 点的最短路径：从 A 点出发，最短路径是前往 C 点；从 C 点出发，最短路径是前往 B 点；从 B 点出发，可以直接前往 D 点。这条路径的长度为 $3+1+2=6$ （千米）。

因城市改建，B 点和 C 点之间的距离增加到了 4 千米，现在请你在更新后的路线图（图 3.2.3）中继续使用贪心策略来规划路线，看是否能够找到距离最短的路线。

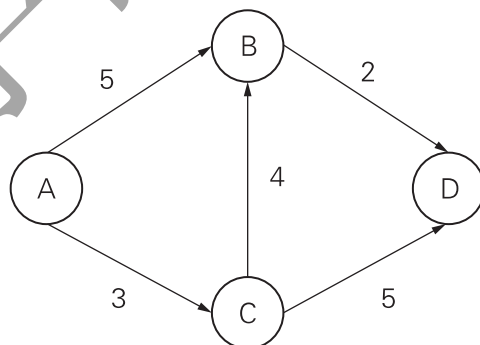


图 3.2.3 路线图

使用贪心策略，得到的路线是：

A→

从 A 点出发，使用贪心策略，在每个节点选择接下来最短的路。

（1）与 A 点相连的两条路中，前往 C 点的路线距离为 3 千米，前往 B 点的路线距离为 5 千米。所以，从 A 点出发后，先前往 C 点，路程为 3 千米。

（2）从 C 点出发，前往 D 点的路线距离为 5 千米，而前往 B 点的路程为 4 千米。所以，从 C 点出发后前往 B 点，路程为 4 千米。

（3）从 B 点出发直接前往 D 点，路程为 2 千米。

这一条线路的总路程为 $3+4+2=9$ （千米），这并不是距离最短的路线。



本路线图中的最短路线为 A—B—D，总路程为 $5+2=7$ （千米）。所以，单纯使用贪心策略无法得到所有最短路径问题的最优解。

三、用狄克斯特拉算法寻找最短路径

想要准确解答所有最短路径问题，我们就要使用更准确的狄克斯特拉（Dijkstra）算法：逐步确定从起始节点到其他节点的最短路径，每次选择当前路径中距离起始节点最近的一个节点，并更新其他节点的距离值。

以图 3.2.4 所示的路线图为例，我们逐步分析狄克斯特拉算法是怎么找到从 A 点到 D 点的最短路径的。在后面的图片中，我们用红色标记已经访问过的节点以及从该节点出发到下一个节点的路线。

第一步：如图 3.2.5 所示，从 A 点出发，在没有访问过的节点中，找到距离最小的节点。如表 3.2.1 所示，当前距离最小的节点是 B 点。下一步以 B 点为途经点，更新没有访问的节点到起点的距离。

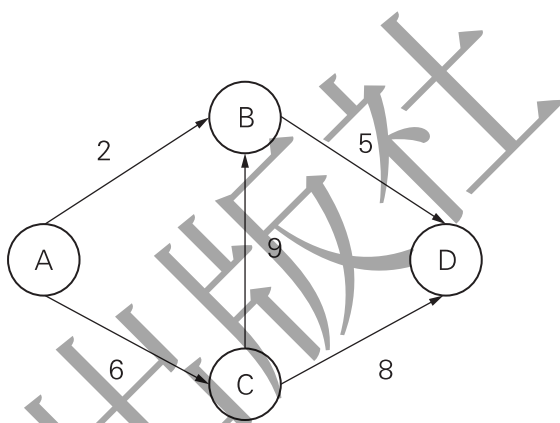


图 3.2.4 路线图

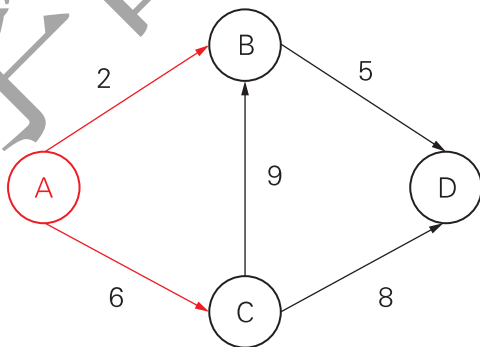


图 3.2.5 A 点出发的路线

表 3.2.1 A 点出发记录

到达节点	B	C	D
距离	2	6	未知
路径	A—B	A—C	A—D
未访问的节点	B, C, D		

第二步：如图 3.2.6 所示，途经 B 点，更新从 A 点出发，途经 B 点前往与 B 点联通的各个节点的距离（表 3.2.2）。在图 3.2.4 所示的有向图中，B 点和 C 点之间只能从 C 点前往 B 点，所以不计算 A 点出发经过 B 点前往 C 点的距离。

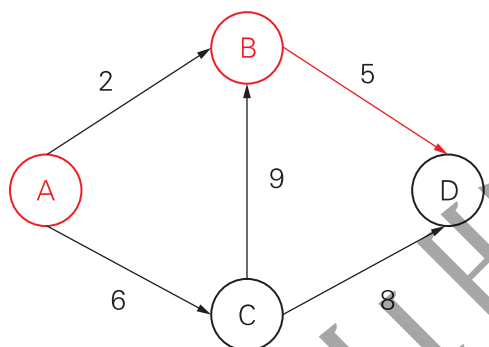


图 3.2.6 途经 B 点的路线

表 3.2.2 途经 B 点记录

到达节点	B	C	D
距离	2	6	7
路径	A—B	A—C	A—B—D
未访问的节点	C, D		

第三步：如图 3.2.7 所示，从 A 点出发，在没有访问过的节点中，找到距离最小的节点，现在是 C 点。以 C 点为途经点，更新没有访问的节点和起点的距离（表 3.2.3）。

途经 C 点到达 B 的距离为 15 千米，比 A 点出发直接到达 B 点的距离更长，所以不更新。

途经 C 点到达 D 点的距离为 14 千米，比 A 点出发经过 B 点到达 D 点的距离更长，所以也不更新。

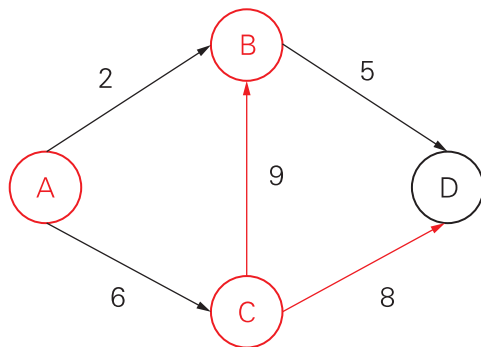


图 3.2.7 途经 C 点的路线



表 3.2.3 途经 C 点记录表

到达节点	B	C	D
距离	2	6	7
路径	A—B	A—C	A—B—D
未访问的节点	D		

第四步：如图 3.2.8 所示，从 A 点出发，在没有访问过的节点中，找到距离最小的节点，现在是 D 点。因为 D 点是最后一个点（终点），在它后面已经不存在未访问的节点了。寻找最短路径的过程结束。

从表 3.2.3 中能看出，从 A 点出发到达 D 点的最短路径为 A—B—D，距离是 7 千米。

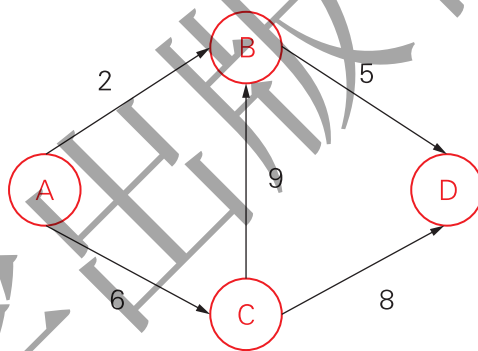


图 3.2.8 途经 D 点的路线

探索

用狄克斯特拉（Dijkstra）算法找出图 3.2.9 中小华到体育馆的最短路径，并在表 3.2.4～表 3.2.6 中填写查找最短路径的过程。

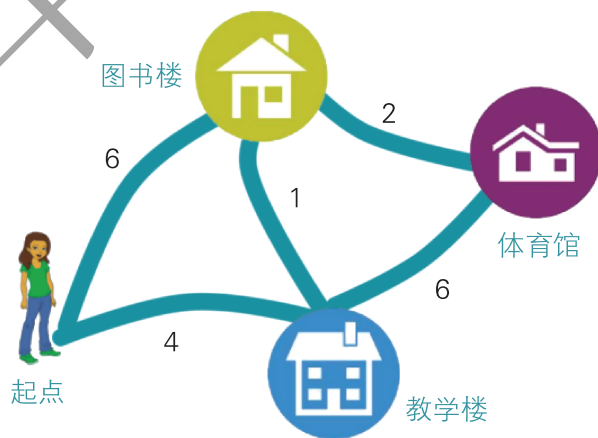


图 3.2.9 小华到体育馆的路线图

表 3.2.4 从 A 点出发记录表

到达节点	B	C	D
距离			
路径			
未访问的节点			

表 3.2.5 途经__点记录表

到达节点	B	C	D
距离			
路径			
未访问的节点			

表 3.2.6 途经__点记录表

到达节点	B	C	D
距离			
路径			
未访问的节点			

挑战

总结提炼狄克斯特拉（Dijkstra）算法解决最短路径问题的 4 个步骤。

第3课

智能识别



本课中你将学习：

- 智能识别在生活中的应用场景
- 计算机分辨物体的原理
- 体验认知 AI

智能识别有着十分广泛的应用，包括机器人、语言识别、图像识别、自然语言处理、专家系统、机器学习、计算机视觉等。本节课，我们将从这些丰富多样的应用实例入手，深入剖析智能识别技术的原理，并利用“认知 AI”设计程序。

一、随处可见的智能识别

随着人工智能技术的发展，智能识别已经得到了广泛的应用，融入了我们生活的方方面面。

1. 图像识别

火车站、机场、工厂、学校、小区等公共场所广泛使用人脸识别技术，可以快速核对人员信息（图 3.3.1），极大地提高了工作效率。在一些移动 App 中也常常使用人脸识别技术来验证是否为本人的操作行为。



图 3.3.1 人脸识别的小区门禁

2. 语音识别

语音识别技术（图 3.3.2）在智能家居领域发挥着关键作用。用户可以通过语音指令控制家中的智能设备，如灯光、空调、电视、音箱等，实现家居环境的智能化和便捷化。例如，用户可以说“打开客厅的灯”，则客厅灯将自动打开。

此外，语音识别技术可以将对话转录为文字，便于智能化处理各种问题。



图 3.3.2 语音识别

3. 文字识别

文字识别能将图片中的文字识别出来，在无人驾驶、智慧交通、智慧停车场、物流零售、身份证识别等场景中得到广泛的运用，极大地提高了工作效率。停车场车牌识别如图 3.3.3 所示。



图 3.3.3 停车场车牌识别

4. 生物识别

指纹识别：很多智能手机采用指纹识别作为解锁方式之一。用户在手机上录入自己的指纹后，每次解锁时，手机通过人工智能指纹识别算法对按压的指纹进行特征提取和比对，只有当比对成功时，才允许用户进入手机系统，提供了一种方便且安全的解锁方式。

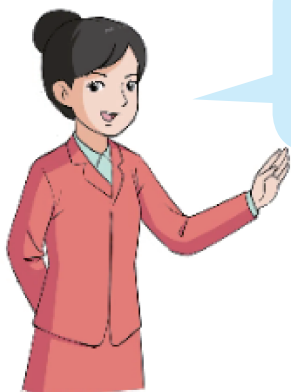
虹膜识别：虹膜识别具有高度的准确性和唯一性，在一些高端安防场景下被广泛应用（图 3.3.4）。例如，在银行金库、重要科研机构等场所，通过虹膜识别来确认人员身份，只有授权人员才能进入，保障了重要场所的安全。



图 3.3.4 虹膜识别技术



二、计算机分辨物体的原理



小青，刚刚我们通过阅读材料了解到了很多智能识别应用的场景。你知道智能设备是怎么识别出信息（如文字）的吗？

老师，我认为智能识别系统会像我们人类一样，先进行学习，明确哪个字长什么样，然后再识别图片上的文字。



智能识别就是利用了机器学习的算法，通过对大量数据进行学习和分析，自动地发现数据里的规律和模式，然后利用这些规律和模式做出准确的判断、预测或者完成各种任务。

我们先来回想一下，如何教小朋友认识草莓和香蕉（表 3.3.1）。

- （1）先让小朋友观察各种草莓和香蕉的外观，比如：形状、颜色等。
- （2）请小朋友总结归纳草莓和香蕉的特征。
- （3）拿出很多品类的水果，请小朋友在里面识别出草莓或者香蕉。

表 3.3.1 小朋友认识草莓和香蕉的过程

种类	观察	归纳特征	识别
草莓		1. 红色且有黑色斑点 2. 形状为圆锥形 3. 顶部有绿色小叶片	比照特征进行识别
香蕉		1. 青绿色，成熟后变黄 2. 形状为弯曲的弓状，有棱 3. 果实间通过果柄连接	

机器学习的过程和小朋友认识水果的过程十分相似，它分为“学习—建模—应用”三个步骤，对应了小朋友认识水果“观察—归纳—识别”的三个阶段。机器学习情绪的过程如表 3.3.2 所示。

表 3.3.2 机器学习情绪的过程

情绪	学习	建模	应用
高兴	提供多种高兴的表情供机器学习 	从很多种表情中提取高兴的面部特征（嘴角上扬、露出牙齿等），建立样本模型	当摄像头或识别框中出现“高兴”或者“伤心”的表情，机器根据样本模型进行判断并识别
伤心	提供多种伤心的表情供机器学习 	从很多种表情中提取伤心的面部特征（嘴角向下，眉头紧皱等），建立样本模型	

三、体验认知 AI

在图形化编程软件中，通过海量表情样本已经训练好的“认知 AI”可识别人脸，获取年龄、情绪、性别、脸型等信息。

打开图形化编程软件，在默认的指令分类中，没有发现“认知 AI”的功能，要从“拓展”中添加“认知 AI”模块（图 3.3.5）。



图 3.3.5 添加“认知 AI”的方法

添加完成后，选项卡上就出现了“认知”选项卡，它包含多个积木（图 3.3.6）。

可以参考图 3.3.7 搭建脚本，了解这些积木块的功能，并将结果填写在表 3.3.3 中。



图 3.3.6 认知模块



图 3.3.7 认知功能测试

表 3.3.3 认知 AI 功能探究

项目	结果
可以识别的情绪	
可以识别的脸型	
能否识别性别	
能否识别年龄	
能否识别佩戴眼镜	

你想用认知AI的功能做一个什么程序呢？

我想做个年龄检测器，有老年人乘公交车的时候自动提醒人们让座。

我想用它做一个开心检测器，我们要多关心不开心的同学。



我想做一个佩戴眼镜检测软件，放在校门口看看有多少同学近视。

我想做一个脸型检测器，为不同脸型做个性服装推荐。

探索

请使用“认知AI”功能设计程序实现拍照识别人脸，并以如图3.3.8所示的方式显示年龄、情绪和脸型。



图 3.3.8 程序运行结果示意图

挑战

用认知AI开发一款情绪检测的小程序，根据检测出的不同情绪，给出不同的激励性的话语。

第 4 课

活动：体验生成式人工智能



本课中你将体验：

- 认识生成式人工智能
- 体验生成式人工智能
- 生成式人工智能的挑战

生成式人工智能（Generative AI）是人工智能的一个分支，是基于算法、模型、规则生成文本、图片、声音、视频、代码等内容的技术。这种技术能够针对用户需求，依托事先训练好的多模态基础大模型等，根据用户输入的相关资料，生成具有一定逻辑性和连贯性的内容。与传统人工智能不同，生成式人工智能不仅能够对输入数据进行处理，更能学习和模拟事物内在规律，自主创造出新的内容（图 3.4.1）。



图 3.4.1 人工智能生成内容（AIGC）



读一读

生成式人工智能主要依赖深度学习模型，通过不断学习数据特征，来生成符合人类对自然性和合理性的要求的新内容。

想象一下，生成式人工智能就像一个“学习力超强的的大脑”。它会“看”成千上万的图片、“听”大量的音乐，或者“读”很多故事，从中学习到它们的样子、声音和结构。当我们给它一个指令，比如“画一只猫”或“写一个小故事”，它就会运用之前学到的内容，创造出一个全新的、独一无二的作品！

生成式人工智能可以帮助我们做很多有趣的事情！

(1) 画画和设计。

它可以根据文字描述画出各种有趣的图片，比如“画一只戴着墨镜的企鹅”（图 3.4.2）或“画一个外星球基地”（图 3.4.3），让我们看到很多天马行空的画作！



图 3.4.2 戴着墨镜的企鹅



图 3.4.3 外星球基地

(2) 写故事和文章。

如果你给它一个主题，它就能根据你的提示写出一个完整的故事，帮助你写作文、写诗，甚至可以写电影剧本（图 3.4.4）。

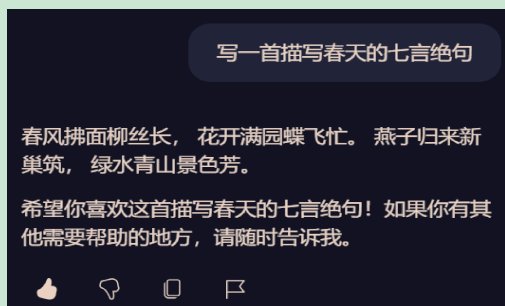


图 3.4.4 生成式人工智能辅助写作



(3) 帮助学习和做题。

生成式人工智能可以帮助我们解答一些问题(图 3.4.5),比如数学题或百科知识问题、计算机编程等,成为学习的小助手。

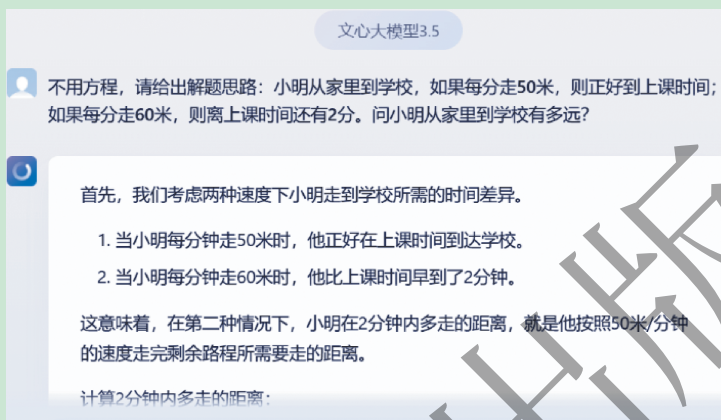


图 3.4.5 生成式人工智能辅助学习

近年来,生成式人工智能如雨后春笋蓬勃发展,国内也产生了许多著名的生成式人工智能网络平台,如文心一言、星火大模型等。

你最想让生成式人工智能做些什么呢?



做一做

打开身边的生成式人工智能的页面,问问 AI:“你能做哪些事情?”,如图 3.4.6 所示。

尝试将你感兴趣的事情让 AI 做一做。



图 3.4.6 向生成式人工智能提问



议一议

原来生成式 AI 会做那么多的事情，你希望它做什么呢？

我想用它快速地梳理一下本学期信息科技课程所学的各种算法。

我想用它帮我翻译图片上的英语短文。

我想用它解答数学课没有听懂的问题。

我想让它帮我规划一个出游攻略。



读一读

生成式人工智能虽然很厉害，但它也存在一些不足。

(1) 可能会出错，生成式人工智能有时会出现奇怪的错误，比如画一只多了几条腿的狗（图 3.4.7），或者写出奇怪的句子。

(2) 需要控制和保护：因为生成式人工智能太强大了，可能有人会用它来制造虚假的图片或信息。因此，我们要学会辨别真伪，不要轻易相信网上的图片和消息。



图 3.4.7 出错的 AI 图片



未来，生成式人工智能会更智能，帮助我们创造出更多有趣的东西，比如虚拟现实游戏、智能机器人伙伴、学习助手等。它会成为我们生活中很好用的小帮手，但也需要我们小心使用哦！

总之，生成式人工智能就像是一个拥有无限创意的“数字画家”和“智慧作家”，我们可以和它一起探索未来的无穷可能！



想一想

有了生成式人工智能，我们还有学习的必要吗？



评一评

同学们，在本课中我们体验了生成式人工智能，请根据表 3.4.1，对学习过程进行评价，在反思中不断成长与进步吧！

表 3.4.1 评价表

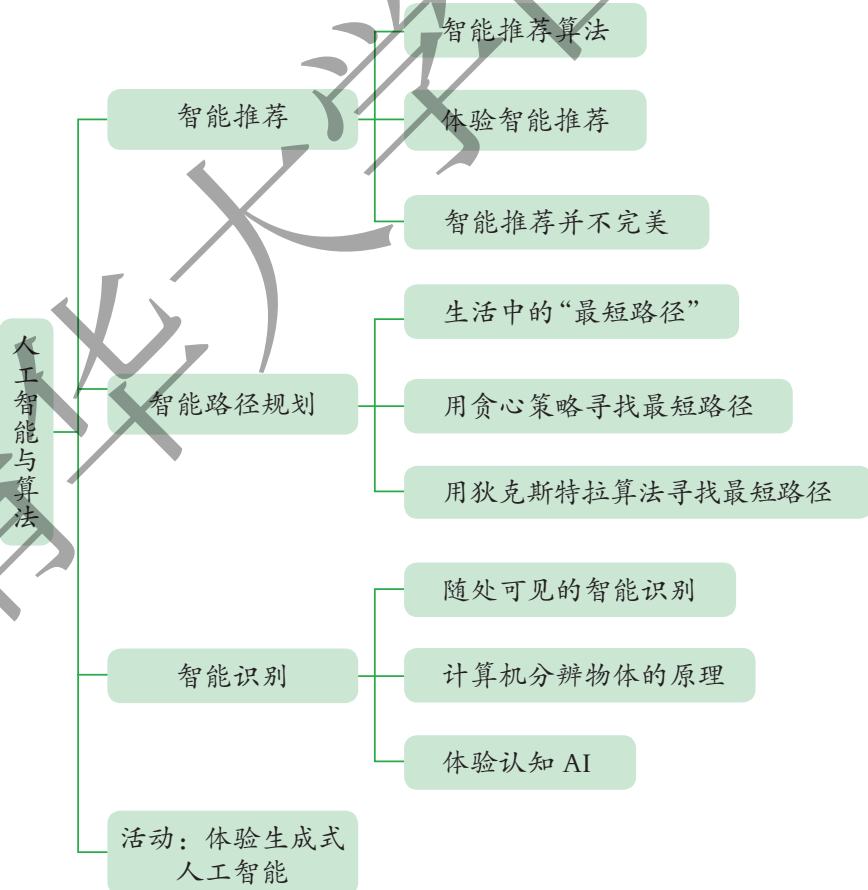
内容	遇到的问题	解决方法	感悟收获	总体评价
读一读				☆☆☆☆☆
做一做				☆☆☆☆☆
议一议				☆☆☆☆☆
读一读				☆☆☆☆☆
想一想				☆☆☆☆☆

小结与评价

本单元你学习了：

- ◆ 智能推荐算法的原理以及不足
- ◆ 智能导航与狄克斯特拉算法
- ◆ 人工智能中的机器学习并体验认知 AI
- ◆ 体验了生成式人工智能

知识梳理





学习检测

1. 为什么总有人对短视频爱不释手浪费了大量时间？你应对电子产品成瘾有哪些方法？

2. 狄克斯特拉算法寻找最短路径与贪心策略有何不同？

3. 认知 AI 为什么能识别面部的情绪表情？

4. 小明完成作业的时候使用生成式人工智能，他的这种做法对吗？我们应该怎样使用生成式人工智能？



反思评价

在本单元学习过程中，肯定少不了与他人互动交流、参与作品制作等活动。现在请就此进行总结与反思，以便更好地促进自身成长。

1. 从同伴那里学到了什么？

2. 向同伴分享了哪些观点？

3. 所运用的工具和方法是否得当？

4. 学习的过程中是否有值得推广的经验？