

《Hello 算法》

动画图解、一键运行的数据结构与算法教程

Stars 74k

Downloads 207k

Contributors 154

靳宇栋 (@krahets) — 著

课程简介

- **定位：** 数据结构与算法入门教程
- **对象：** 具有基础编程技能的人士
- **重要性：** 计算机学科的地基与纽带

课程内容

- **复杂度分析：** 迭代与递归、时间复杂度、空间复杂度
- **数据结构：** 数组、链表、栈、队列、哈希表、树、堆、图
- **算法：** 搜索、排序、分治、回溯、动态规划、贪心

学习方式

9.2 图基础操作

删除顶点: 在邻接矩阵中删除一行一列。当删除首行时达到最差情况, 需要将 $(n-1)^2$ 个元素“向上移动”, 从而使用 $O(n^2)$ 时间。

初始化: 传入 n 个顶点, 初始化长度为 n 的顶点列表 vertices, 使用 $O(n)$ 时间; 初始化 $n \times n$ 大小的邻接矩阵 adjmat, 使用 $O(n^2)$ 时间。

图

邻接矩阵

图 9-7 邻接矩阵的初始化、增删边、增删顶点

以下是基于邻接矩阵表示图的实现代码:

Python C++ Java C# Go Swift JS TS Dart Rust C Zig

graph_adjacency_matrix.py

```

my_heap.py
109 """Driver Code"""
110 if __name__ == "__main__":
111     # 初始化大顶堆
112     max_heap = MaxHeap([9, 8, 6, 6, 7, 5, 2, 1, 4, 3, 6, 2])
113     print("\n输入列表并建堆后")
114     max_heap.print()
115
116     # 获取堆顶元素
117     peek = max_heap.peek()
118     print(f"\n堆顶元素为 {peek}")
119
PORTS  GITLENS  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  ...  zsh  +  ...  ...  ^  ...  ^  X
~/Dow/hello-algo main > | Py base 18:18:47
  
```

3.1 数据结构分类

Hello 算法

第 0 章 前言

第 1 章 初识算法

第 2 章 复杂度分析

第 3 章 数据结构

3.1 数据结构分类

3.2 基本数据类型

3.3 数字编码

3.4 字符编码

3.5 小结

第 4 章 数组与链表

第 5 章 栈与队列

第 6 章 哈希表

第 7 章 树

第 8 章 堆

第 9 章 图

第 10 章 搜索

第 11 章 排序

第 12 章 分治

第 13 章 回溯

第 14 章 动态规划

第 15 章 贪心

第 16 章 附录

参考文献

目录

3.1 逻辑结构: 线性与非线性

3.1.2 物理结构: 连续与分散

请问: 线性数据结构和非线性数据结构, 哈希表为什么都存在。

我觉得应该是因为哈希表在没有冲突之前, 是用数组存储的, 当出现哈希冲突之后, 有各种解决冲突的方式, 比如链地址法, 所以两种逻辑结构都包括它, 不像其他的数据结构都是只用数组或者只用链表实现的

好问题~这因为哈希表底层是数组, 所以包含线性数据结构, 而了解决哈希冲突, 我们可能会使用“链地址” (后续列表章节会讲), 在链地址中, 哈希表的每个桶 (索引) 都存储一个链表; 当链表长度超过一定阈值时, 又可能被转化为树 (常为红黑树), 以提升查询效率。因此, 哈希表可能同时包含线性 (数组、链表) 和非线性 (树) 数据结构。

输入 预览 Aa

取消 回答

动画图解

「点亮地图」

一键运行

「动手实践」

互助学习

「知识内化」

风格约定

- 字体大小适合横屏观看
- 代码使用 Python ， 配有简洁的注释

```
"""标题注释，用于标注函数、类、测试样例等"""
```

```
# 内容注释，用于详解代码
```

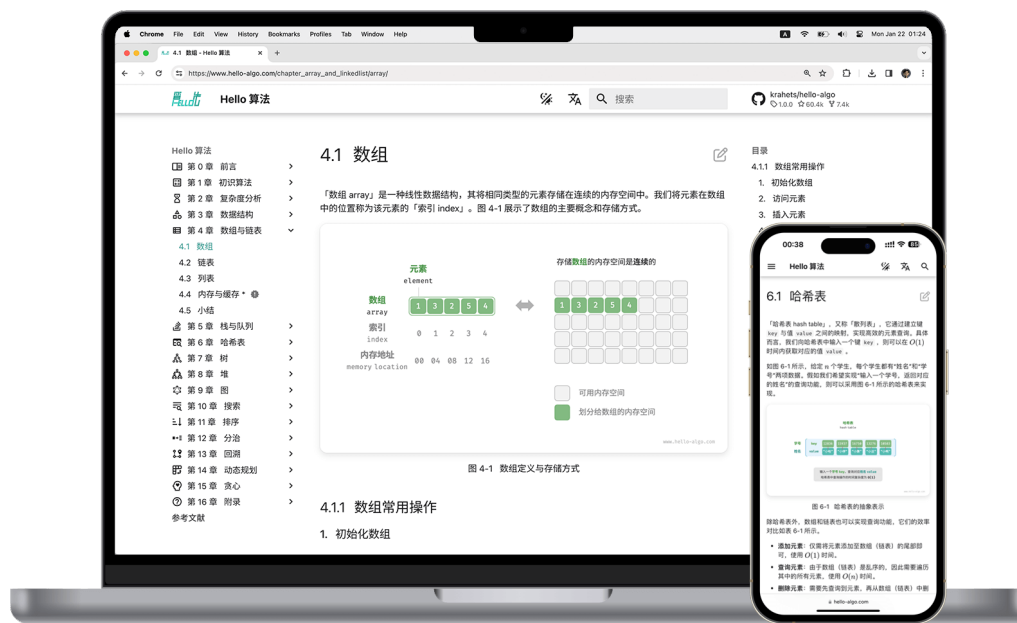
- 标题后标注 * 的是选读章节
- 专业术语会添加英文翻译与下划线，例如 数组 (array)
- 重点内容会**加粗**，有特指含义的词句会用“引号”标注



配套教材



纸质书

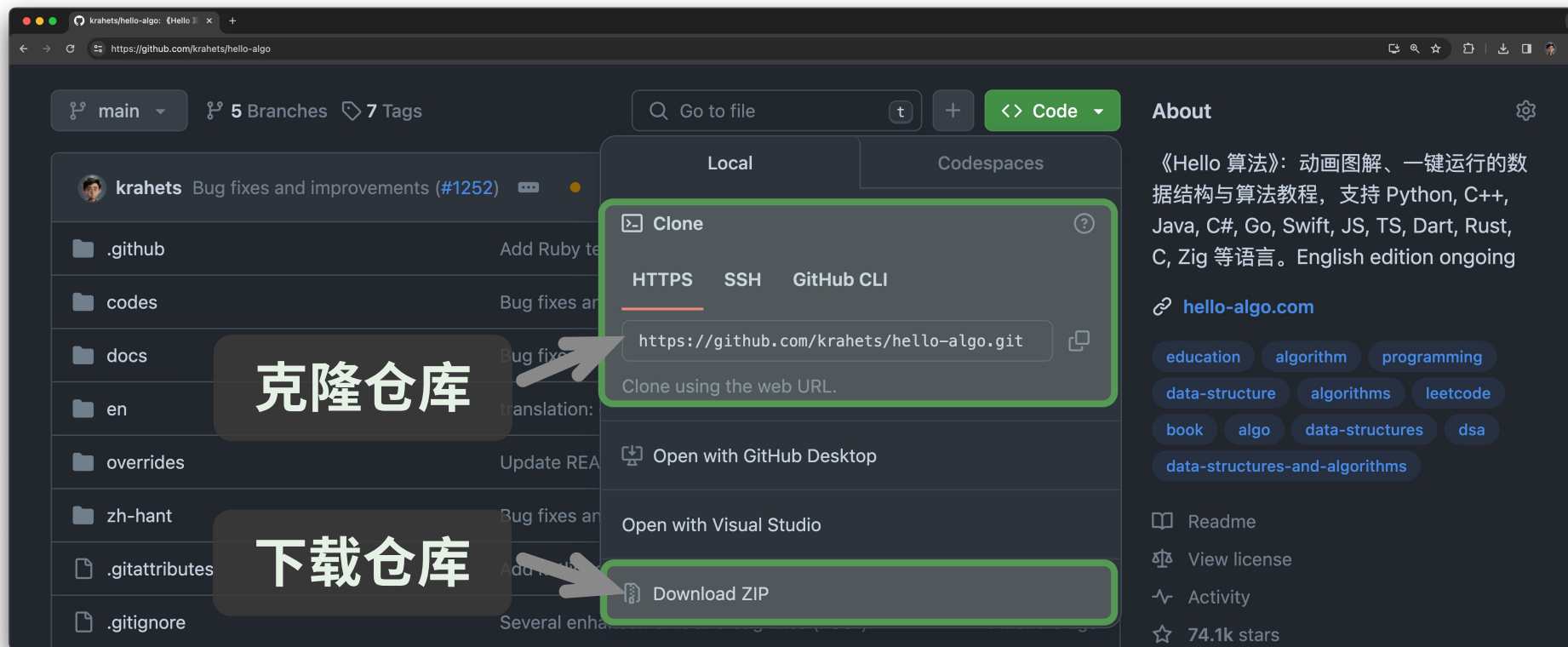


网页版



PDF 版

配套代码



github.com/krahets/hello-algo

