

动态规划算法

创作者：
时间：2024年X月

目录

- 第1章 简介
- 第2章 一维动态规划
- 第3章 二维动态规划
- 第4章 多阶段动态规划
- 第5章 动态规划优化技巧
- 第6章 总结

• 01

第一章 简介

动态规划算法概述

动态规划算法是一种解决复杂问题的方法。通过拆分问题，将问题分成小问题并逐步解决。动态规划算法通常用于解决具有重叠子问题和最优子结构性质的问题。

动态规划算法特点

具有最优子结构性质：一个问题的最优解可以通过其子问题的最优解来推导。具有重叠子问题性质：子问题之间存在重叠，可以通过存储中间结果来避免重复计算。递归和迭代两种实现方式。

01 贪心算法

每一步选择当前状态下最优解，不能回溯

02 动态规划

考虑所有可能的选择，找到最优解

03

动态规划的适用场景

最优解依赖

解决问题的最优解
依赖于其子问题的
最优解

子问题拆分

问题可以被拆分成
子问题，并且每个
子问题的最优解可
以推导出整体问题
的最优解

避免重复计算

可通过存储中间结
果来避免重复计算

动态规划算法特点

动态规划算法具有最优子结构性质和重叠子问题性质，递归和迭代两种实现方式。

动态规划算法实例

Fibonacci数列

使用动态规划算法
计算

0-1背包问题

动态规划的经典应
用之一

最长公共子序 列

动态规划解决字符
串相关问题

最大子序列和

利用动态规划求解

动态规划与传统算法对比

动态规划

适用于有重叠子问题和最优子结构性质的问题

需要考虑所有可能的选择

通过存储中间结果来避免重复计算

传统算法

没有考虑子问题之间的关联
无法解决具有最优子结构性质的问题

不能处理重叠子问题

动态规划优势

能够解决一些传统算法无法解决的问题

可以提高算法效率

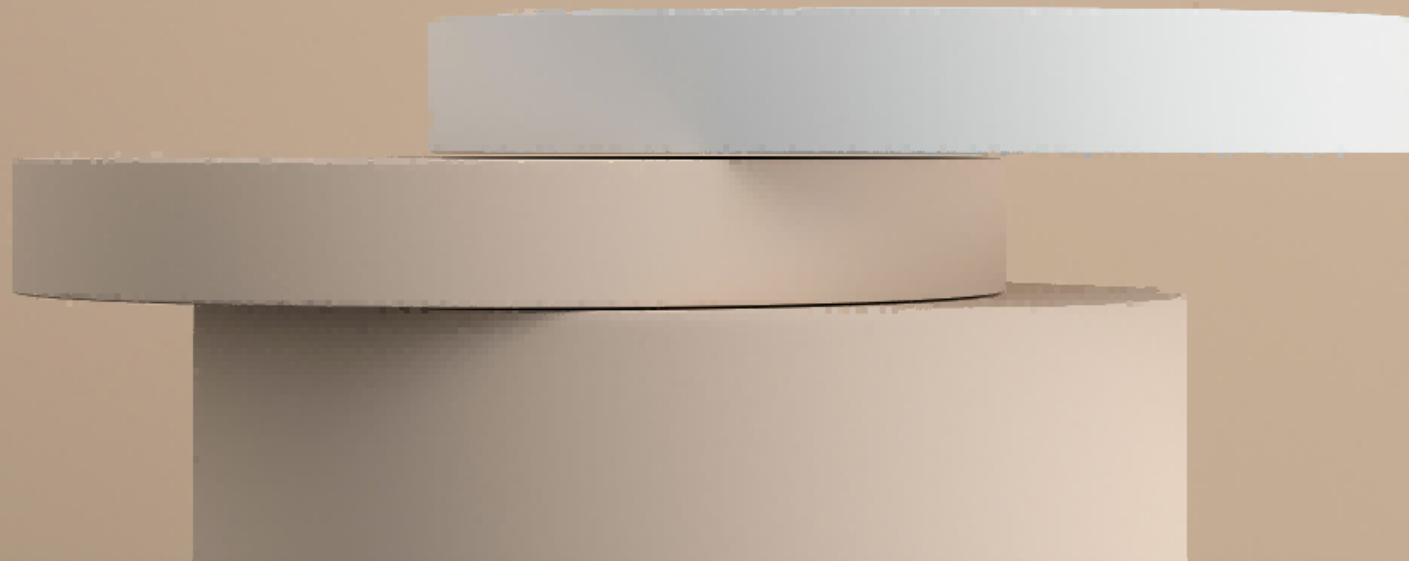
适用于复杂问题的求解

应用范围

动态规划适用于很多实际问题的解决

在计算机科学领域有广泛应用

能够处理一些复杂的优化问题



01 确定状态

定义问题的状态，并找到状态转移方程

02 初始化

初始化动态规划数组或者表格

03 状态转移

根据状态转移方程来推导出最优解

• 02

第2章 一维动态规划

斐波那契数列

斐波那契数列是一个经典的动态规划问题，可以通过递归、迭代和动态规划三种方法来解决。其中，动态规划方法的空间复杂度优于递归方法，更适合处理大规模数据。

背包问题

0-1背包问题

只能选择一次

多重背包问题

每种物品有限次选择

完全背包问题

可重复选择

01 动态规划求解

利用动态规划数组存储长度

02 子序列长度

以当前元素结尾

03 优化算法

减小时间复杂度

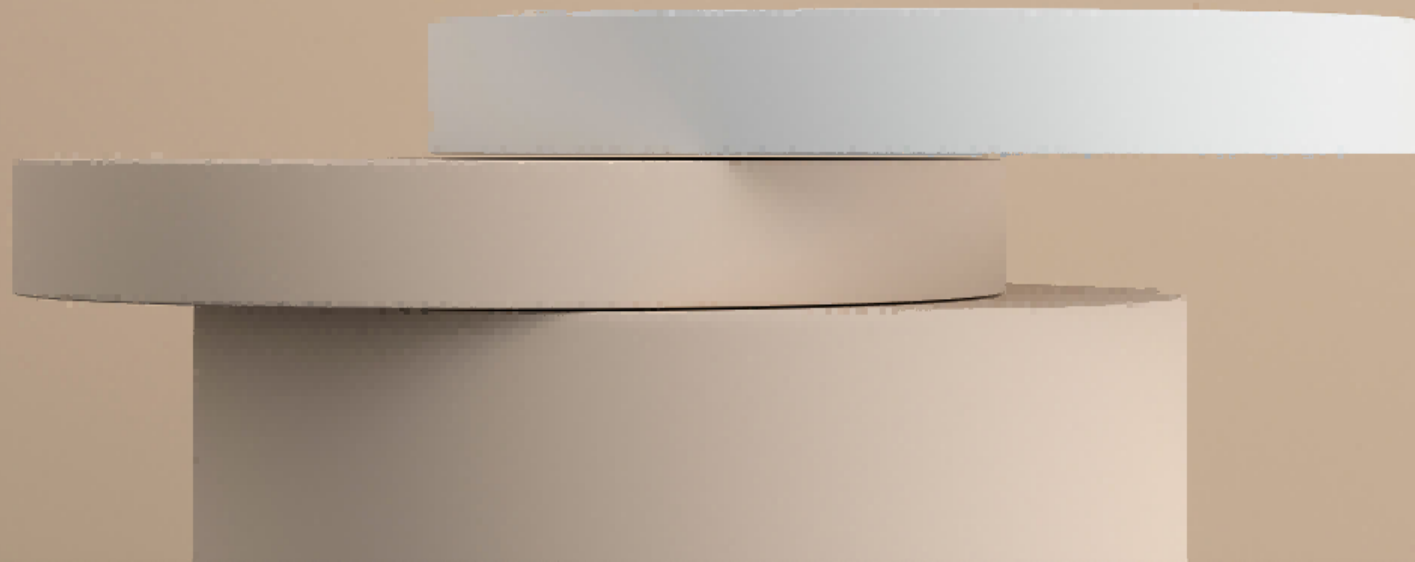
爬楼梯问题

动态规划求解

转化为斐波那契数列

方法数计算

求解爬n级楼梯的方法数



总结

一维动态规划是算法设计中常用的方法，可应用于各种问题的解决。掌握动态规划算法可以提高问题求解的效率和准确性，是程序员必备的技能之一。继续学习更多动态规划问题，加深理解，拓展思维。

• 03

第3章 二维动态规划

矩阵路径问题

矩阵路径问题是一个经典的二维动态规划应用。通过定义一个二维数组来保存到达每个位置的最短路径和，可以有效地解决这个问题。动态规划的思想是利用之前计算的结果来简化当前的计算。

矩阵路径问题解决方法

定义二维数组

保存最短路径和

遍历整个矩阵

计算最短路径

获取终点值

即为最短路径和

从起点开始

逐步更新路径值

最大正方形问题解决方法

定义二维数组

保存最大正方形边
长

更新最大边长

取最大值

时间复杂度

$O(m*n)$

遍历矩阵

计算正方形边长

字符串编辑距离

字符串编辑距离是一个常见的动态规划问题。通过定义一个二维数组来保存转化为目标字符串的最小操作次数，可以有效地解决字符串之间的相似度计算。动态规划的核心是找到最优子结构和重叠子问题，以提高计算效率。

字符串编辑距离解决方法

定义二维数组

保存最小操作次数

递推计算

更新最小操作次数

获取目标值

即为最小操作次数

初始化边界值

空字符串操作次数

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/938125024077006056>