

预处理数组优化

前置知识

讲解066 ~ 讲解088 - 【必备】阶段动态规划专题，整个专题都需要

动态规划优化专题的讲述顺序

专题1：优化尝试和状态设计-上，讲解127

专题2：优化尝试和状态设计-下，讲解128

专题3：倍增优化，讲解129

专题4：单调队列/单调栈优化，讲解130

专题5：树状数组/线段树优化，讲解131

专题6：预处理数组优化，讲解132，本节

以上内容是【扩展】课程阶段，动态规划优化的讲述内容

单调性优化的更多内容，比如斜率优化、四边形不等式优化等，会在【挺难】阶段的视频中讲述

预处理数组优化

题目1

休息 k 分钟最大会议和

给定一个长度为 n 的数组 arr ，表示从早到晚发生的会议，各自召开的分钟数

当选择一个会议并参加之后，必须休息 k 分钟

返回能参加的会议时长最大累加和

比如， $arr = \{ 200, 5, 6, 14, 7, 300 \}$ ， $k = 15$

最好的选择为，选择200分钟的会议，然后必须休息15分钟

那么接下来的5分钟、6分钟、14分钟的会议注定错过

然后放弃7分钟的会议，而选择参加300分钟的会议

最终返回500

$1 \leq n$ 、 $arr[i]$ 、 $k \leq 10^6$

来自真实大厂笔试，对数器验证

利用预处理数组减少枚举的可能性

预处理数组优化

题目2

炮兵阵地

给定一个 $n * m$ 的二维数组 *grid*，其中的 **1** 代表可以摆放炮兵，**0** 代表不可以摆放
任何炮兵攻击范围是一个"十字型"的区域，具体是上下左右两个格子的区域
你的目的是在 *grid* 里摆尽量多的炮兵，但要保证任何两个炮兵之间无法互相攻击
返回最多能摆几个炮兵

$1 \leq n \leq 100$

$1 \leq m \leq 10$

$0 \leq grid[i][j] \leq 1$

测试链接：<https://www.luogu.com.cn/problem/P2704>

利用预处理数组减少枚举的可能性

预处理数组优化

题目3

还原数组的方法数

原本有一个长度为 n 的数组 arr ，下标从1开始，数组中都是 ≤ 200 的正数

并且任意 i 位置的数字都满足： $arr[i] \leq \max(arr[i-1], arr[i+1])$

特别的， $arr[1] \leq arr[2]$ ， $arr[n] \leq arr[n-1]$

但是输入的 arr 中有些数字丢失了，丢失的数字用0表示

返回还原成不违规的 arr 有多少种方法，答案很大需要对998244353取余

$3 \leq n \leq 10^4$

测试链接：<https://www.nowcoder.com/practice/49c5284278974cbda474ec13d8bd86a9>

不减少枚举的可能性，而是利用预处理数组加速枚举的过程

预处理数组优化

题目4

粉刷房子III

房子有 n 个，从左到右排列，编号 $1..n$ ，颜色有 c 种，编号 $1..c$

给定数组 $house$ ， $house[i]$ 表示房子的颜色，如果 $house[i]$ 为0说明房子没有涂色

你必须给每个没有涂色的房子涂上颜色，如果有颜色的房子不能改变颜色

给定二维数组 $cost$ ， $cost[i][v]$ 表示如果 i 号房涂成 v 号颜色，需要花费多少钱

相邻的、拥有同一种颜色的房子为1个街区

比如如果所有房子的颜色为: $\{1, 1, 2, 3, 2, 2\}$ ，那么一共4个街区

最终所有的房子涂完颜色，一定要形成 t 个街区，返回最少的花费

$1 \leq t \leq n \leq 100$ $1 \leq c \leq 20$

$0 \leq house[i] \leq c$ $1 \leq cost[i][v] \leq 10^4$

测试链接：<https://leetcode.cn/problems/paint-house-iii/>

不减少枚举的可能性，而是利用预处理数组加速枚举的过程

预处理数组优化

题目5

从上到下挖砖块

一堆组成倒三角形形状的砖埋在地里，一共有 n 层，第1层有 n 块砖，每层递减，类似如下数据

4 2 9 5

3 1 7

2 4

8

需要从第1层开始挖，每挖开一块砖都可以获得数值作为收益，第1层的砖可以随意挖

但是挖到下面的砖是有条件的，想挖 i 层的 j 号砖，你需要确保 $i-1$ 层的 $(j, j+1)$ 块砖已经被挖开

最多可以挖 m 块砖，返回最大的收益

$1 \leq n \leq 50$ $1 \leq m \leq 1300$ 砖块数值 ≤ 100

测试链接：<https://www.luogu.com.cn/problem/P1437>

不减少枚举的可能性，而是利用预处理数组加速枚举的过程