

树状数组原理、扩展、代码详解

前置知识

讲解047、讲解048 - 一维差分、二维前缀和 & 二维差分

本节课讲述：

树状数组维护信息的类型 & 树状数组与线段树的比较 & 线段树预告

一维数组上实现：单点增加、范围查询的树状数组

一维数组上实现：范围增加、单点查询的树状数组

一维数组上实现：范围增加、范围查询的树状数组

二维数组上实现：单点增加、范围查询的树状数组

二维数组上实现：范围增加、范围查询的树状数组

下节课讲述：

树状数组相关题目，进一步练习树状数组的使用

树状数组原理、扩展、代码详解

树状数组维护信息的类型

树状数组一般用来维护可差分的信息

比如：累加和、累乘积、或者出题人发现了某个可差分信息来出题考你

不可差分的信息，比如：最大值、最小值、除此之外的很多信息

不可差分的信息一般不用树状数组维护，会选择用线段树维护，因为线段树维护的方式思考难度更低

树状数组维护不可差分信息的内容不再讲述，后续会详细讲解线段树

大多数情况下，线段树可以替代树状数组，两者的时间复杂度差不多，单次调用都是 $O(\log n)$

线段树的优势：用法全面、思考难度低、维护信息类型多（包括可差分信息、不可差分信息）

线段树的劣势：代码较多、使用空间较大、常数时间稍差

树状数组优势：代码量少、使用空间少、常数时间优异

树状数组劣势：维护信息的类型少、维护某些不可差分的信息时思考难度大并且不易实现

线段树会从讲解110开始讲述

树状数组原理、扩展、代码详解

题目1

树状数组单点增加、范围查询模版

测试链接：<https://www.luogu.com.cn/problem/P3374>

这是使用树状数组最常见的方式，笔试、比赛中都大量出现
树状数组中信息的组织方式，[课上重点图解](#)

当 i 位置增加 v 时，如下位置都获得该值： $i += i \& -i$

当计算 $[1..i]$ 范围的累加和时，把如下位置的值都加上： $i -= i \& -i$

神奇的`lowbit`，可以将上述的组织，优雅、方便的实现

以上所有细节，[课上都重点图解](#)

单次调用时间复杂度 $O(\log n)$

树状数组原理、扩展、代码详解

题目2

树状数组范围增加、单点查询模版

测试链接：<https://www.luogu.com.cn/problem/P3368>

理解这个用法需要先理解一维差分，请先掌握讲解047的视频，前13分钟的内容

理解了上述内容，针对本用法，课上会重点图解

单次调用时间复杂度 $O(\log n)$

这种用法往往会被线段树替代，不过树状数组实现代码很少，很容易写

树状数组原理、扩展、代码详解

题目3

树状数组范围增加、范围查询模版

测试链接：<https://www.luogu.com.cn/problem/P3372>

理解这个用法需要先理解一维差分，请先掌握讲解047的视频，前13分钟的内容

理解了上述内容，针对本用法，需要进一步分解一维差分，课上图解一下

单次调用时间复杂度 $O(\log n)$

这种用法其实是线段树的模版，但是树状数组也能实现，而且代码少

不过一般遇到这种问题就选择线段树实现了

树状数组原理、扩展、代码详解

题目4

二维数组上单点增加、范围查询，使用树状数组的模版

测试链接：<https://leetcode.com/problems/range-sum-query-2d-mutable/>

理解这个用法需要先理解二维前缀和，请先掌握讲解048的视频，前1小时06分的内容

本用法和题目1类似，一维数组上构建树状数组，只在一个维度上展开，本用法在两个维度上展开即可

证明比较复杂，记住即可，好在非常容易记忆，代码也简单

单次调用时间复杂度 $O(\log n * \log m)$

二维数组上范围增加、单点查询的用法，不再讲述，因为二维数组的这些用法可以被题目5一并解决

树状数组原理、扩展、代码详解

题目5

二维数组上范围增加、范围查询，使用树状数组的模版

测试链接：<https://www.luogu.com.cn/problem/P4514>

理解这个用法需要先理解二维前缀和 & 二维差分，请先掌握讲解048的视频，前1小时06分的内容

本用法需要进一步拆解二维差分，课上图解一下

单次调用时间复杂度 $O(\log n * \log m)$

这个用法的代码实现，java实现和C++实现的说明

该用法也可以用线段树的树套树替代，替代之后使用空间更大，改写的代码更多，理解会更难

线段树会从讲解110开始讲述，但是线段树的树套树会在【挺难】标签下的视频里讲述