

数据结构

Data Structure

AHdoc

11/08/2017



Recall

- 最简单的数据结构
 - 队列，栈，链表（双向链表），HASH
- 稍微复杂一些的简单数据结构
 - 堆，排序二叉树，平衡树
- 最重要的数据结构
 - 线段树（静态数组），Splay（动态数组）
- 和图有关的数据结构
 - 并查集
 - 树：树表示，树分治（点分治，链分治）
- More practical
 - 多个数据结构同时使用
 - 数据结构的嵌套
 - 分块

今日话题

- 1, 线段树
 - 1.1, 线段树要维护什么?
 - 1.2, 线段树 与 树
 - 1.3, 超大数组 与 动态数组
- 2, 数据结构的嵌套
 - 2.1, 线段树 套 数组
 - 2.2, 一般的结构嵌套



1，线段树

- 线段树是最适合思维训练的数据结构
- 几乎所有别的数据结构都与之相关
- 两个“**ACM**游戏”中最基础的思维考点：
 - 动态规划（思维难点：对逻辑问题量化）
 - 线段树（思维难点：如何维护信息）

1.1, 线段树要维护什么?

- 维护什么?
 - 维护缺少什么
- 三个重点:
 - 在一个区间上（对应线段树一个结点）的多次区间操作，可以合并为同一个区间操作。
 - 一个区间上的操作可以分解为对左右2个区间的操作。
 - 可以快速合并两个区间的信息。

1.1, 线段树要维护什么?

- 维护什么?
 - 维护缺少什么
- 最简单的构造线段树的方法:
 - Step 1, 维护所有询问
 - Step 2,
 - 对于所有修改: 考虑这个修改对所有询问的影响。把每次修改后新的询问答案与原答案的“差”也维护下来 (如果需要维护新的东西, 重新考察Step 2)
 - 对于合并: 考虑每个修改对合并区间的影响。把每次修改后大区间的值与原答案的“差”也维护下来 (如果需要维护新的东西, 重新考察Step 2)

1.1, 线段树要维护什么?

— 例1

— 输入一个长为 n 的数列，维护 m 个操作，操作分为三类：

- (1) 某连续段一起加上一个常数
- (2) 询问某一段的所有数的两两乘积的和
- (3) 询问某一段的所有相邻两数乘积的和

1.1, 线段树要维护什么?

— 例2

- 输入两个长都为 n 的数列
 - $A[1], A[2], A[3], \dots, A[n]$
 - $B[1], B[2], B[3], \dots, B[n]$
- 维护 m 个操作, 操作分为三类:
 - (1)把 $A[l, r]$ 同时加上一个常数
 - (2)把 $B[l, r]$ 同时加上一个常数
 - (3)对于所有 x in $[l, r]$, $A[x] += B[x]$
 - (4)对于所有 y in $[l, r]$, $B[y] += A[y]$
 - (5)询问: A 的区间平方和; B 的区间平方和
 - (6)询问: 一个区间 $[l, r]$ 内 $A[x] * B[x]$ 的和

1.1, 线段树要维护什么?

— 例3

— 输入长度为 n 的数列 $a[i]$ 。维护 m 次操作，每次操作可以：

- (1) $a[l] \dots a[r]$ 每一项都加一个常数 C
- (2) 求 $F[a[l]] + F[a[l]+1] + \dots + F[a[r]]$
- (3) 求 $F[a[l]] + F[a[l+1]] + \dots + F[a[r]]$

— 其中 $F[i]$ 表示斐波那契数列。即 $F[0]=F[1]=1$,
 $F[n+2]=F[n+1]+F[n]$ 。 ($C \leq 10^{11}$)

1.1, 线段树要维护什么?

— 例4

- 给定初始数列 $A[1], A[2], A[3], \dots, A[n]$
- 操作:
 - $[L, R] [s, t]$ (保证 $R-L=t-s$)
 - 将 $A[L, R]$ 修改为斐波那契数列 第 s 到 第 t 项
- 询问:
 - 区间和

1.1, 线段树要维护什么?

—例5

- 有一个 $6*n$ 的点阵，平行于坐标轴的方向上相邻的点之间可以连边（带权），维护以下操作：
 - (1)在某两点之间连边（若可以连边的话）
 - (2)拆除某条边
 - (3)询问某两点是否连通，和最短路径长度

1.1, 线段树要维护什么？ —例6

- 给定数列 $A[1], A[2], \dots, A[n]$
- 修改：
 - 区间加一
 - 区间减一
- 询问：

$$\left\{ \sum_{L < i < R} F[A_{i-1} + 1] F[A_{i+1} - 1] \right\} \bmod (10^9 + 7)$$

1.2, 线段树 与 树

- 看询问：
 - 1, 子树 - DFS序
 - 2, BFS序下若干层 - BFS序
 - 3, 两点之间路径 - 拆分路径, 倍增, 树链剖分
 - 4, 同时询问路径与子树 - 树链剖分
 - 5, 询问一个点相邻的所有点 - 逐点维护
 - 6, 询问路径, 子树, 一个点相邻点 - 树链剖分

1.2, 线段树 与 树一例1

- 输入一个 n 个点的边有权的**无根树**。维护 m 个操作，操作共有如下两类：
 - (1)修改某条边一侧的子树上的点权，同时加一
 - (2)询问单点点权
 - (3)询问某条边一侧的子树上的点权和

1.2, 线段树 与 树一例2

- 输入一个 n 个点的边有权的有根树。维护 m 个操作，操作共有如下两类：
 - 修改：
 - 单点修改权值
 - 修改某个点为根的子树内所有权值（加一个常数）
 - 询问：
 - 单点询问
 - 询问到一个点距离为1的所有点权值和
 - 询问到一个点距离为2的所有点权值和
 - 询问到一个点距离为3的所有点权值和

1.2, 线段树 与 树一例3

- 输入一个 n 个点的边有权的有根树。维护 m 个操作，操作共有如下两类：
 - 修改：
 - 将某路径上的点权加上一个常数
 - 询问：
 - 给定 u, v, k 询问 u 到 v 的路径上所有到 u 距离是 k 的倍数的点的权值和

1.2, 线段树 与 树一例4

- 输入一个 n 个点的边有权的有根树。维护 m 个操作，操作共有如下两类：
 - (1)将某子树中的所有边权都增加一个常数 C
 - (2)求某两点之间路径上的边权的平方和

1.2, 线段树 与 树一例5

- 输入一个 n 个点的边有权的**无根树**。维护 m 个操作，操作共有如下两类：
 - 修改
 - 修改一条路径上的权值，加一个常数
 - 询问
 - 询问一个点附近一圈点的权值和（距离为1的点）

1.2, 线段树 与 树一例6

- 很粗的树上路径问题
 - 有 n 个带权无向图 G_1, G_2, \dots, G_n
 - $|V(G_i)| \leq 3$
 - n 个“无向图”形成一棵树（每一个图看做一个结点）
 - 只有树上相邻2个结点（无向图）才会共享“点”
 - 每一个“点”出现在的 结点 形成了一颗子树
 - 求 支持路径修改 的 路径询问

1.2, 线段树 与 树一例6

- 很粗的树上路径问题
 - 树 T —图 G
- 树链剖分后的链
 - 对应到一维看是什么？
 - 区间（对应树上的路径）只需要考虑左右结点内的“点”
 - 维护左右结点内的“点”的可达性

1.3, 超大数组 与 动态数组

- 线段树 是 维护 **静态数组** 的动态修改的
- 数组范围 $[1, n]$
- 实际上可以做到任意大的范围
 - 建立超大的线段树 $[1, L]$
 - 如果一个区间信息太匮乏（或者没有信息）可以直接存储为单点
 - 而不需要维护整个树

1.3, 超大数组 与 动态数组

- 线段树 是 维护 **静态数组** 的动态修改的
- 实际上也可以维护动态数组
 - 如何动态:
 - 区间插入区间删除
 - 区间反转
 - 区间交换
 - 区间快速copy (成多份)
 - Splay:
 - 依然可以做标记
 - 如何做区间修改和区间查询:
 - » 去遍历所有包含的子区间? 错!
 - » 把考察的区间splay到根附近, 在单子树上操作

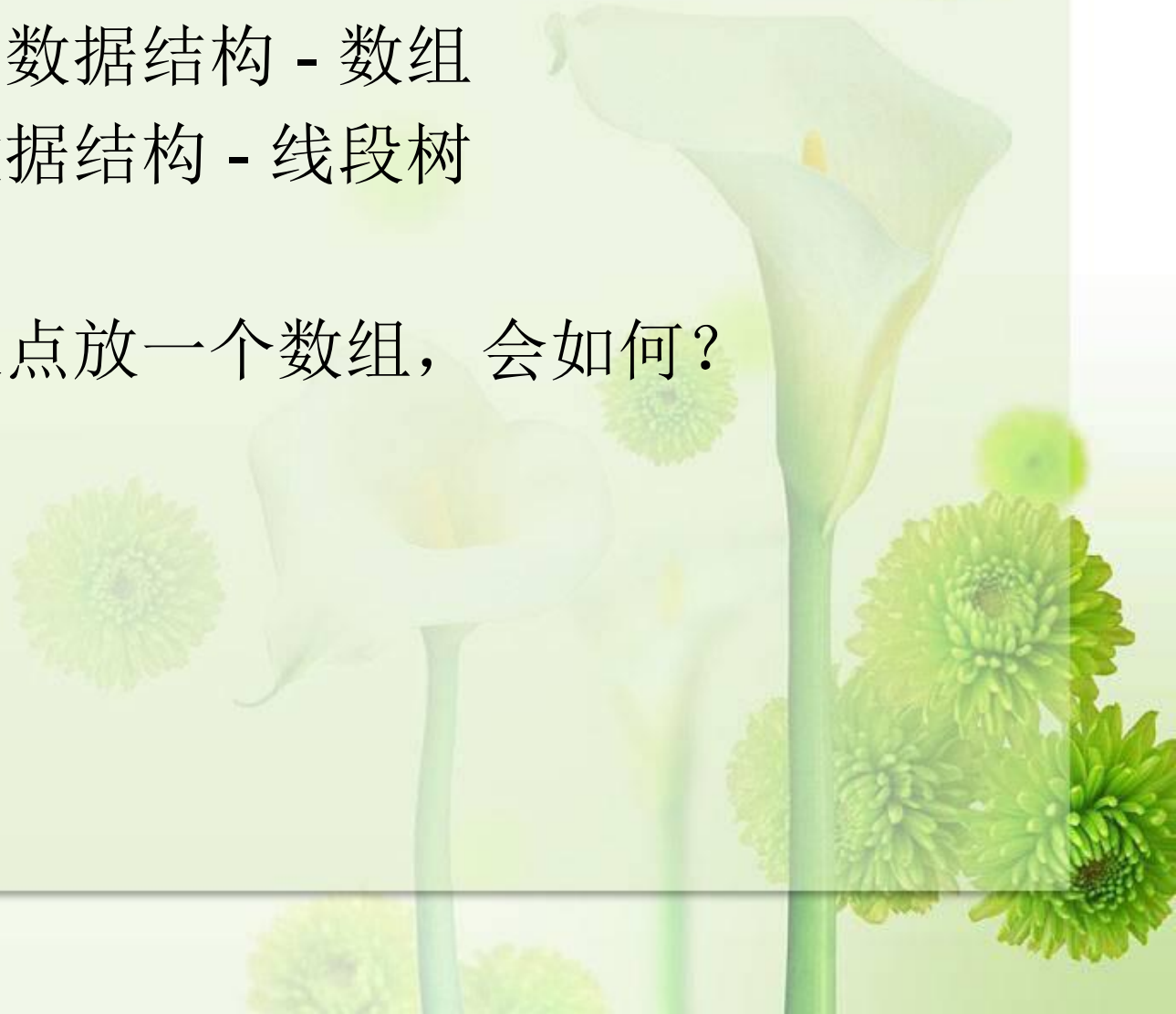
2， 数据结构的嵌套

- 2.1) 线段树套数组
- 2.2) 一般的结构嵌套



2.1 线段树套数组

- 作为最简单的数据结构 - 数组
- 和最基础的数据结构 - 线段树
- 如果线段树逐点放一个数组，会如何？



2.1 线段树套数组

- Level 0: $[a_1, a_2, a_3, a_4, a_5, a_6, \dots]$
- Level 1: $[b_1, b_2, b_3, \dots]$ $[c_1, c_2, c_3, \dots]$
- Level 2: $[\dots]$ $[\dots]$ $[\dots]$ $[\dots]$
- 一般来说：每一个结点嵌套的数组可能不一样大
- 直观感觉：根的数组应该要比子节点的大
- 主要考察相邻2个结点的数组之间是什么关系

2.1 线段上套数组

- 小区间 可以 是大区间的一种压缩
- [day1,day2,day3,...,day7,day8,day9,...,day14,day15]
- 2016-
- [doc(day1-7),gap(day8),doc(day9-11),xiaodao(day12-15)]
- 2017-
- [claris(day1-7),doc(day8-14),home(day15)]

2.1 线段上套数组

- 小区间 是大区间的一种压缩
- 大区间（根）的某连续一段 对应 小区间的一个点
- 这实际上就是函数式线段树
- 根：数组与时间轴对应 (time 1 to time n)
- 子节点维护的数组：
 - 对于不变化的连续时间戳，存储一个点
 - 根的数组中每一个点 指向 子节点数组中的一个位置

2.1 线段上套数组

- 大区间（根）的某连续一段 对应 小区间的一个点
- 这实际上就是函数式线段树
- 一个更加暴力的思考方式：
 - 每一个结点维护一个区间
 - 每一个区间维护所有时间节点的情况
 - 压缩时间节点（相同状态的时间结点被合并）
 - 每次在一个区间上二分找到考察的时间结点
 - （时间复杂度比之前多了一个 \log ）
 - （不需要任何相邻区间之间的关联信息）

一般的结构嵌套

- 再来想一个简单的数据结构
- 既然数组是数据结构，那么树当然也是
- 线段树 套 树！



一般的结构嵌套

- 线段树 套 树!
- 动态的子树查询修改问题
- 给定 n 个叶子结点（分别带有权），看作是 n 个独立的树
- 修改：
 - 选择两个树，合并
 - （新建一个根结点，两个树变成根的左右子树）
 - 修改以某个结点为根的子树内所有叶子结点的权值
 - 加一个常数
 - 乘一个常数
- 询问
 - 询问某个点为根的子树内结点的权值和

今日训练题

- 线段树，需要多练习！
- 请完成Claris讲义和习题中的所有题目
- 做题不在多，但是不能做水题
- hdu 5891
- hdu 6087
- 欢迎提问！