

主要内容

- 口一,分块
 - 静态分块
 - 动态分块
 - 高维的情况
- □二,数据结构的嵌套
 - 一个最简单的例子
 - 线段树作为外层数据结构



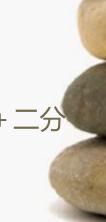
静态分块 Successor

■ Sean owns a company and he is the BOSS. The other Staff has one Superior.every staff has a loyalty and ability. Some times Sean will fire one staff. Then one of the fired man's Subordinates will replace him whose ability is higher than him and has the highest loyalty for company. Sean want to know who will replace the fired man.



静态分块 Successor

- □ A[i] , B[i] ;
- □ 询问: [L,R] 中 A大于c的位置中B最大的一个?
- □暴力:
 - 排序
 - 二分(关于c)
 - 维护前缀最大值
- □分块
 - 块:排序(预处理)+前缀最大值(预处理)+二分
 - 其它:暴力(最多只有两块)



动态分块

- □ 需要保证的因素:
 - 每一块大小不小于 aK (K)
 - 每一块大小不大于等于 bK (2K)
- □操作:
 - 插入
 - 删除
 - ×翻转
 - ×交换
 - ×整段删除
 - ×移动



动态分块 KeyInsertion

□ N个士兵,从左至右有M个位置

- □将军下达一个命令Goto(L,S):
 - 若L为空,士兵S站在L上
 - 若L上有士兵K,士兵S站在L上,执行Goto(L+1,K)
 - 每个士兵被下达命令一次且仅一次
 - 命令执行过程中,士兵站的位置标号超过M
- □要你求出最后队列的状态



高维的情况

- 口分块个数
- □每一块大小
- □每一块独立处理的时间



高维的情况 酗酒者

- □N²的格点地图,每一个路口相邻的四条边有
 - -p(u,left) + p(u,right) + p(u,up) + p(u,down) = 1
- □Q次询问:
 - (x1,y1)与(x2,y2)
 - 随机游走
 - 期望用时



高维的情况 酗酒者

□相关的问题:

□格点图 vs 平面图?

- □高斯消元?
- □矩阵的特性?
- □快速高斯消元?



数据结构的嵌套

- □1)一个最简单的例子
 - 分块(不同块之间互不影响)+数组
 - 时间轴的维护
 - 可持久化结构的雏形
- □2)线段树作为外层数据结构
 - 再看"最简单的例子"
 - 其它推广



一个最简单的例子 Grid surveillance

Big Brother needs a new machine that will keep track of all the people moving into town.

Problem specification

The town is a grid G[0..4095][0..4095]. Initially, all entries in G are zeroes. The machine will need to support two types of instructions: additions and queries. Each addition modifies a single cell of the grid. Each query asks you about the sum of a rectangle. However, queries can also ask about older states of the grid, not only about the present one.

In order to have a smaller input file (and also in order to force you to process the instructions in the given order) we used the input format given below. There is a helper variable c, initially set to zero. Let $\varphi(c,x) = (x \text{ xor } c) \text{ mod } 4096$. The instructions are processed as follows:

- Each addition is described by three integers x, y, a. To process it, first compute the new coordinates $x' = \varphi(c,x)$ and $y' = \varphi(c,y)$. Then, add a to the cell G[x'][y']. Finally, set c to the current value of G[x'][y'].
- Each query is described by five integers x_1 , x_2 , y_1 , y_2 , t. To process it, first compute the new coordinates $x_i' = \varphi(c, x_i)$ and $y_i' = \varphi(c, y_i)$. If necessary, swap them so that $x_1' \le x_2'$ and $y_1' \le y_2'$.

Next, take the grid G_t that is defined as follows: If t = 0, then $G_t = G$. If t > 0, G_t is the state of G after the very first t additions. (If t exceeds the total number of additions so far, $G_t = G$.) Finally, if t < 0, then G_t is the state of G before the very last -t additions. (If -t exceeds the total number of additions so far, G_t is the initial state.)

The answer to the query is the sum of all $G_t[i][j]$ where $x_1' \le i \le x_2'$ and $y_1' \le j \le y_2'$. This sum is also stored in c.

一个最简单的例子 Grid surveillance

- □ (1)如何分块,如何处理独立的块
 - 二维前缀和(预处理) O(1)
 - -64×64
- □ (2)时间轴与每一个块
 - multi-fold blocks!
 - The elements won't be numbers, but arrays of pairs.
 - Each pair will represent an update as the timestamp and the new value.
 - 总共需要维护的块
- □ (3)如何支持询问 每一个multi-fold blocks:二分
- □ (4)时间复杂度分析
 - O(nqlogq)



一个最简单的例子 Grid surveillance

□思考:

- 无法针对多个块同时关于时间轴分治!
- 换成线段树?



线段树作为外层数据结构

- □每一个结点:
 - 对应一个区间
 - 维护数组 (pair<time,val>)
- □相邻的结点:
 - 两个区间
 - 对应关系?(依赖时间)
- □ 找到关于时间戳的所有位置:
 - 逐个二分?
 - 利用相邻结点的线序性!
 - 均摊O(1)



再看最简单的例子 Grid surveillance

P[16]

P[8]

P[4]

P[12]

P[2]

P[6]

P[10]

P[14]

A[1] A[2] A[3] A[4] A[5] A[6] A[7] A[8] A[9] A[10] A[11] A[12] A[13] A[14] A[15] A[16]

- □ (1) 不考虑时间:
 - All of these queries run in O(lg²n) time.
- (2)节点: multi-fold, 数组维护 pair<time,val>
- □ (3)根位置:二分
- □ (4)其余位置:利用指针关系
- □ 时间复杂度:O(lg²n)-对比直接二分每一个节点:O(lg³n)



线段树作为外层数据结构 其他推广

- □1)平衡树
 - 区间k值
 - 启发式合并
- 口2)堆
 - 只存储在叶子节点
- □3)线段树
 - 维护多维数据



今日训练

- **□** HDU4286
- **□** BZOJ3110
- **□** BZOJ1507
- **□** HDU4729
- **□** HDU4866
- □ ipsc.ksp.sk/2011/real/problems/g.html
- ☐ ipsc.ksp.sk/train/submit
- □ 欢迎提问

