

济南市历城第二中学信息学奥林匹克竞赛校内模拟赛

STL 的应用实例

yL0I2021 模拟赛

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	可持久化完全动态图上树状数组维护 01 背包转移	在基尔霍夫矩阵里反演三次后用矩阵树定理求欧拉函数	在圆方树上用可追溯化实链剖分查询两个九十九维多面体的交	签到题
英文题目与子目录名	hard	sohard	veryhard	difficult
可执行文件名	hard.exe	sohard.exe	veryhard.exe	difficult.exe
输入文件名	hard.in	sohard.in	veryhard.in	difficult.in
输出文件名	hard.out	sohard.out	veryhard.out	difficult.out
每个测试点时限	1s	2s	4s	1s
捆绑测试	否	是	否	是
附加样例文件	有	有	有	有
结果比较方式	全文比较 (过滤行末空格及文末回车)			
题目类型	传统	传统	传统	传统
运行内存上限	512M	512M	512M	512M

二. 提交源程序文件名

对于 C++ 语言	hard.cpp	sohard.cpp	veryhard.cpp	difficult.cpp
-----------	----------	------------	--------------	---------------

三. 编译开关 (开启 -O2 优化, 以 C++11 标准进行编译)

对于 C++ 语言	-lm -Wl,--stack=512000000 -std=c++11 -O2
-----------	--

四. 注意事项:

- 1、禁止粘贴赛前已写过的任何代码, 即你所提交的所有代码必须是赛时完成的。
- 2、文件名 (程序名和输入输出文件名) 必须使用英文小写。
- 3、C++ 中函数 `main()` 的返回值类型必须是 `int`, 程序正常结束时的返回值必须是 `0`。
- 4、统一评测时采用的机器配置为: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 内存 16GB。上述时限以此配置为准。
- 5、只提供 Linux 格式附加样例文件。
- 6、在评测时, 数据均以在 Windows 系统下以「输出为二进制文件」形式生成, 也即换行符均为 `\n` (ASCII 值为 10)。选手文件夹下所提供的附加样例文件也是以该形式生成的。如果你不知道本条的含义, 你可以理解为: 你可以将输入数据当作在 Linux 下正常生成的而无需对换行符进行任何特殊处理。
- 7、评测机在相同编译开关下将 10^6 个范围在 $[0, 2^{64})$ 中均匀生成的随机整数插入 `std::set` 的用时为 800ms。
- 8、评测时选手程序可用的栈空间大小与题目空间限制一致, 且计入程序所用空间。
- 9、必须在 21:15 以前提交代码, 请合理安排时间。

1. 可持久化完全动态图上树状数组维护 01 背包转移

(hard.cpp)

【题目背景】

“你这样写题目名字是要向全国人民谢罪的。” N 老师对 zxy 哔哔说。

【问题描述】

众所周知，洛谷有一道题叫《普通平衡树》，是洛谷最著名的高级数据结构模板。但普通平衡树太普通了，不符合银临女神出众的气质，因此扶苏希望你写一棵《特殊平衡树》。

您需要写一个**可重集**，来维护一些数，其中需要提供以下操作：

1. 插入一个值为 x 的数。
2. 删除一个值为 x 的数(如果没有这个数，则不删除)。
3. 求集合中最大的数(如果集合为空，输出 zay)。
4. 求集合中最小的数(如果集合为空，输出 zay)。
5. 求 x 的前驱(前驱定义为**小于** x ，且最大的数；如果没有前驱，输出 cuc)。
6. 求 x 的后继(后继定义为**大于** x ，且最小的数；如果没有后继，输出 cyc)。
7. 将目前集合中的所有数字都加上 x 。

可重集是一个集合，但是集合中的元素可以重复。例如， $\{1, 1, 4, 5, 1, 4\}$ 是一个可重集； $\{1, 2\}$ 也是一个可重集；向可重集 $\{1, 2\}$ 插入一个数字 2 后的集合为 $\{1, 2, 2\}$ 。

【输入格式】

输入文件名为 `hard.in`。

输入的第一行是一个整数 n ，表示操作的条数。

为了体现本题的特殊性，本题采用强制在线的方式读入数据。具体来说，你需要维护一个整数 $lastans$ ，这个数表示你上次输出的答案，初始时为 0。每次进行输出时，如果输出的不是一个字符串，则你需要将 $lastans$ 赋值为你本次的输出。

接下来 n 行，每行一个或两个整数，表示一次操作。第一个数为 opt 。 $opt \otimes lastans$ 的值为表示本次操作的编号。其中 \otimes 表示按位异或运算符，即 C++ 语言的 \wedge 运算符。

- 若编号为 3 或 4，则该行只有一个数字，表示一次查询。
- 否则接下来有一个整数，表示给定的 x 。

【输出格式】

输出文件名为 `hard.out`。

对于每次查询操作，输出一行一个整数表示答案。

【输入输出样例 1】

hard.in	hard.out
5	114514
1 1	
1 1	
1 2	
7 114513	
5 114515	

见选手目录下的 `hard/hard1.in` 和 `hard/hard1.ans`。

【输入输出样例 2】

见选手目录下的 `hard/hard2.in` 和 `hard/hard2.ans`。
这个样例满足「只有前六个操作」。

【输入输出样例 2】

见选手目录下的 `hard/hard3.in` 和 `hard/hard3.ans`。

【数据规模与约定】

本题共有 20 个测试点，每个测试点 5 分。

测试点编号	n=	特殊性质
1	114514	没有 3, 4, 5, 6 操作
2~6	1001	无
7~9	100002	只有前四个操作
10~12	100003	只有前六个操作
13~16	100004	没有 5 和 6 操作
17~20	100005	没有特殊性质

对于全部的测试点，保证 $\text{opt} \otimes \text{lastans}$ 的值是 $[1, 7]$ 范围内的整数； $0 \leq x < 2^{63}$ ；任意时刻集合中的数都是在 $[0, 2^{63}-1]$ 范围内的整数；对于 5、6 两个操作，不保证给出的 x 存在于集合中。

【提示】

n 的数值可以快速帮你判断测试点所具有的特殊性质。

2. 在基尔霍夫矩阵里反演三次后用矩阵树定理求欧拉函数

(sohard.cpp)

【题目背景】

“你这样写题目名字是要向全国人民谢罪的。” zxy 哔哔对 zay 咕咕说。

【问题描述】

南有仙地，名曰摩梭，摩梭有湖，泸沽是也。

茶茶在泸沽湖中寻找自己的梦。氤氲雾气中，茶茶的 n 个梦排成了一个序列。茶茶的所有梦境都是拉瓦的样子。为了区分这些拉瓦，茶茶规定从左到右第 i 个的拉瓦的美颜值是一个非负整数 a_i 。面对着这些梦，茶茶会进行 m 次操作，每次操作会给定两个数字 p, x ，然后将 a_p 和 a_{p+1} 都对 x 做按位异或。每次操作完之后，茶茶都想知道，当前的梦序列中，有多少个子区间 $[l, r]$ ，满足 $1 \leq r$ 且区间的异或和为 0，请你回答茶茶的问题。

区间 $[l, r]$ 的异或和定义为 $a_l \otimes a_{l+1} \otimes \cdots \otimes a_{r-1} \otimes a_r$ 。其中 \otimes 代表二进制按位异或运算，即 C++ 语言的「^」运算符。两个区间不同当且仅当两区间左端点不同或两区间右端点不同或两区间左右端点均不同。

为了避免输出过大，你只需要输出四个整数，分别表示你所有回答的按位异或之和、你共有多少次回答的答案是奇数，你的所有答案中的最大值、你的所有答案中的最小值。

【输入格式】

输入文件名为 sohard.in。

第一行有两个整数，分别表示梦境的数量 n 和操作的次数 m 。

第二行有 n 个用空格隔开的整数，第 i 个整数表示第 i 个拉瓦的美颜值 a_i 。

接下来 m 行，每行两个整数，依次表示一次操作的 p 和 x 。

【输出格式】

输出文件名为 sohard.out。

输出四行，每行一个整数，依次表示你所有回答的按位异或之和、你共有多少次回答的答案是奇数，你的所有答案中的最大值、你的所有答案中的最小值。

【输入输出样例 1】

sohard.in	sohard.out
5 3	3
1 2 3 4 5	3
1 3	3
2 3	1
3 3	

见选手目录下的 sohard/sohard1.in 和 sohard/sohard1.ans。

【输入输出样例 2】

见选手目录下的 sohard/sohard2.in 和 sohard/sohard2.ans。

【数据规模与约定】

本题采用多测试点捆绑测试，共有 6 个子任务。

- 子任务 1（10 分）：保证 $n, m \leq 100$ 。
- 子任务 2（10 分）：保证 $n, m \leq 300$ 。
- 子任务 3（20 分）：保证 $n, m \leq 3000$ 。
- 子任务 4（30 分）：保证 $n, m \leq 10^5$ 。
- 子任务 5（30 分）：无特殊约定。

对于前四个子任务，保证 $a_i, x \leq n$ ；

对于全部的测试点，保证 $1 \leq n, m \leq 10^6$ ， $0 \leq a_i, x \leq 10^9$ ， $1 \leq p < n$ 。

【说明】

不能通过本题的选手将得到《来自凉凉的鼓励》。

3. 在圆方树上用可追溯化实链剖分查询两个九十九维多面体的交

(veryhard.cpp)

【题目背景】

“你这样写题目名字是要上烤架的。”zay 咕咕对 zqm 嘎嘎说。

【问题描述】

Sylvia 是一个热爱学习的女孩子。

前段时间, Sylvia 参加了学校的军训。众所周知, 军训的时候需要站方阵。Sylvia 所在的方阵中有 $n \times m$ 名学生, 方阵的行数为 n , 列数为 m 。

为了便于管理, 教官在训练开始时, 按照从前到后, 从左到右的顺序给方阵中的学生从 1 到 $n \times m$ 编上了号码 (参见后面的样例)。即: 初始时, 第 i 行第 j 列的学生的编号是 $(i-1) \times m + j$ 。

然而在练习方阵的时候, 经常会有学生因为各种各样的事情需要离队。在一天中, 一共发生了 q 件这样的离队事件。每一次离队事件可以用数对 (x, y) ($1 \leq x \leq n, 1 \leq y \leq m$) 描述, 表示第 x 行第 y 列的学生离队。

在有学生离队后, 队伍中出现了一个空位。为了队伍的整齐, 教官会依次下达这样的两条指令:

1. 向左看齐。这时第一列保持不动, 所有学生向左填补空缺。不难发现在这条指令之后, 空位在第 x 行第 m 列。

2. 向前看齐。这时第一行保持不动, 所有学生向前填补空缺。不难发现在这条指令之后, 空位在第 n 行第 m 列。

教官规定不能有两个或更多学生同时离队。即在前一个离队的学生归队之后, 下一个学生才能离队。因此在每一个离队的学生要归队时, 队伍中有且仅有第 n 行第 m 列一个空位, 这时这个学生会自然地填补到这个位置。

因为站方阵真的很无聊, 所以 Sylvia 想要计算每一次离队事件中, 离队的同学的编号是多少。

注意: 每一个同学的编号不会随着离队事件的发生而改变, 在发生离队事件后方阵中同学的编号可能是乱序的。

【输入格式】

输入文件名为 veryhard.in。

输入共 $q+1$ 行。

第 1 行包含 3 个用空格分隔的正整数 n, m, q , 表示方阵大小是 n 行 m 列, 一共发生了 q 次事件。

接下来 q 行按照事件发生顺序描述了 q 件事件。每一行是两个整数 x, y , 用一个空格分隔, 表示这个离队事件中离队的学生当时排在第 x 行第 y 列。

【输出格式】

输出文件名为 veryhard.out。

按照事件输入的顺序, 每一个事件输出一行一个整数, 表示这个离队事件中离队学生的编号。

【输入输出样例 1】

veryhard.in	veryhard.out
2 2 3	1
1 1	1
2 2	4
1 2	

见选手目录下的 veryhard/veryhard1.in 和 veryhard/veryhard1.ans。

【输入输出样例 1 解释】

$$\begin{array}{l}
 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} \\
 \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} \\
 \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 1 \\ 3 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 1 \\ 3 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}
 \end{array}$$

列队的过程如左图所示，每一行描述了一个事件。

在第一个事件中，编号为 1 的同学离队，这时空位在第一行第一列。接着所有同学向左标齐，这时编号为 2 的同学向左移动一步，空位移动到第一行第二列。然后所有同学向上标齐，这时编号为 4 的同学向上一步，这时空位移动到第二行第二列。最后编号为 1 的同学返回填补到空位中。

【输入输出样例 2】

见选手目录下的 veryhard/veryhard2.in 和 veryhard/veryhard2.ans。

【数据规模与约定】

测试点编号	n	m	q	其他约定
1~6	≤1000		≤500	无
7~10	≤50000			
11, 12	=1	≤100000		所有事件 x=1
13, 14		≤300000		
15, 16	≤300000			
17, 18	≤100000			无
19, 20	≤300000			

对于全部的测试点，保证 $1 \leq x \leq n$, $1 \leq y \leq m$ 。

【提示】

请注意常数因子对程序效率造成的影响。如果有人的正解在复杂度合理的前提下被卡常，评测时会酌情开大时限。

4. 签到题

(difficult.cpp)

【题目背景】

这是一道q(ian)d(ao)t(i)。当然，也许是一道q(ue)d(e)t(i)。

【问题描述】

因为这套题是 12 月 16 日扶苏生病回家晚上赶出来的，所以扶苏并没有精力给这个题写有(du)趣(liu)的题面。

给你一个长度为 n 的序列，和 q 条操作。

序列初始全部为 0， q 条操作每条有三个参数 l, r, x ，意为给序列 $[l, r]$ 区间内的每个数加上 x 。

现在，对于 $1 \leq k \leq n$ ，求有多少个 k 满足存在一种选择的方式，使得从 q 条操作中选出若干条操作（操作不必要连续），只进行这些操作后序列的**最大值**为 k 。对于每个 k ，一条操作只能进行一次；两个不同的 k 之间操作互不影响。

如果你认为题意不够清晰，可以结合样例进行理解。

【输入格式】

输入文件名为 difficult.in。

第一行有两个整数，分别表示序列的长度 n 和操作条数 q 。

接下来 q 行，每行三个整数，分别表示依次操作的 l, r, x 。

【输出格式】

输出文件名为 difficult.out。

输出一行若干个整数，按**从小到大**的顺序输出所有符合要求的 k 。

【输入输出样例 1】

difficult.in	difficult.out
4 3 1 3 1 2 4 2 3 4 4	1 2 3 4

见选手目录下的 difficult/difficult1.in 和 difficult/difficult1.ans。

【输入输出样例 1 解释】

选择操作 1 可以使序列的最大值为 1；选择操作 2 可以使序列的最大值为 2；选择操作 1、2 可以使序列的最大值为 3（操作后的序列为 $\{1, 3, 3, 2\}$ ）；选择操作 3 可以使序列最大值为 4。

【输入输出样例 2】

difficult.in	difficult.out
7 2 1 5 1 3 7 2	1 2 3

见选手目录下的 difficult/difficult2.in 和 difficult/difficult2.ans。

【输入输出样例 3】

difficult.in	difficult.out
10 3 1 1 2 1 1 3 1 1 6	2 3 5 6 8 9

见选手目录下的 difficult/difficult3.in 和 difficult/difficult3.ans。

【输入输出样例 4】

见选手目录下的 difficult/difficult4.in 和 difficult/difficult4.ans。

【数据规模与约定】

本题采用多测试点捆绑测试，共有 4 个子任务。

- 子任务 1（5 分）：为第一个样例。
- 子任务 2（20 分）：保证 $n, q \leq 10$ 。
- 子任务 3（35 分）：保证 $q \leq 10$ 。
- 子任务 4（40 分）：无特殊约定。

对于全部的测试点，保证 $1 \leq n, q \leq 1000$ ， $1 \leq l, r, x \leq n$ 。