

关于 Splay 树，可以先看：

<http://blog.csdn.net/leolin/article/details/6436037>

我的模板代码基本上是参考了

<http://blog.csdn.net/whyorwhnt/article/details/10416739> 这个页面中下方的链表版代码，并且有些许改动。Splay 是一种无比强大的数据结构，基本上线段树能干的它都能干，线段树不能干的它也能干。我用一个模板代码来应对各种题，所以代码量有些大。

P0J3580 【中等】

题目大意：

要求实现一种数据结构，支持对一个数字序列的 6 种操作：

ADD $x\ y\ val$ ：第 x 个数到第 y 个数之间的数每个加 D ；

REVERSE $x\ y$ ：第 x 个数到第 y 个数之间全部数翻转；

REVOLVE $x\ y\ c$ ：第 x 个数到第 y 个数之间的数，向后循环流动 c 次，即后面 c 个数变成这段子序列的最前面 c 个，前面的被挤到后面。

INSERT $x\ P$ ：在第 x 个数后面插入一个数 P 。

DELETE x ：删除第 x 个数。

MIN $x\ y$ ：求第 x 个数到第 y 个数之间的最小数字。

输入：

第一行有一个整数 N ，表示有 N ($1 \leq N \leq 100000$) 个数字。下面有一行，有 N 个整数，用空格分开，依次表示序列的第 1 到第 N 个数。然后有一行，包含一个整数 M ，表示有 M ($1 \leq M \leq 100000$) 个操作。下面有 M 行，每一行有一条操作指令，格式如上述。

输出：

每一条 MIN 操作输出一行，包含一个整数，即所求区间的最小值。

题解：

这题在 Splay 的题目里不是最简单的，但是是最适合拿来写模板的。首先有一个很重要的操作：提取区间，要操作 $[a, b]$ 区间，那么将 $a-1$ 先旋转到根节点，再将 $b+1$ 旋转到根节点的右子节点，那么由中序遍历的性质可以知道，此时根节点的右子节点的左子树是我们要操作的区间。因此在提取了操作的区间以后：

1、对于区间加的操作，只需要加上一个加标记，并且修改这个节点维护的区间最值信息；

2、对于区间翻转操作，只需要加上一个翻转标记；

3、对于区间插入操作，如果插入到 pos，那么我们可以选择区间 [pos+1, pos]（对，大的在前）也就是把 pos 旋到根，把 pos+1 旋到根的右子结点，并且此时根的右子结点的左子树是空的，然后插入到根的右子结点的左子结点上（这个操作也可以把砍下来的一棵子树重新拼接上去）；

4、对于删除结点操作，只需将该结点放入回收队列中，并且修改根的右子结点的信息即可（砍树的时候只需要断开父子关系并且修改父亲结点信息）；

5、对于求区间最值操作，由于每一个节点本身已经维护了一个子树信息，直接提取即可；

6、对于区间移位操作，在右移之后，可以发现 [a, b] 被分为两个区间 [a, b-c] [b-c+1, b]，将后者插入到前者之前即可（c 可能很大，需要先取模）。所有操作除了翻转以外都需要先把延迟标记推下来，否则就会更新不及时而出错，具体见代码。

另外，由于内存占用比较多，所以我加上了一个内存回收队列，把删掉的节点放入到那里去，然后插入新节点的时候先看回收队列中是否有可用节点，没有的话再向内存池申请。

HDOJ1754 【基础】

题目大意、输入输出要求见原题的中文。

题解：

查询操作已经分析过了，讲讲修改操作。有一种偷懒的做法（修改第 k 个学生）：查询 [k, k] 的最小值，然后给 [k, k] 区间加上差值即可。

HDOJ3487 【基础】

题目大意：

给出一个数字序列，要求支持两种操作：

- 1、CUT a b c，表示把序列中的 [a, b] 区间切下来，放到切完以后的第 c 个数后面；
- 2、FLIP a b，表示翻转 [a, b] 区间。

输入：

有若干组测试数据。每一组测试数据第一行有两个整数 N 和 M，表示有 N 个（ $1 \leq N \leq 300000$ ）整数组成一个序列，这个序列里的数依次是 1 到 N，然后有 M 个（ $1 \leq M \leq 300000$ ）操作。N=M=-1 时测试数据结束。接下来有 M 行，每一条操作占一行，格式如上所述。

输出：

每一组测试数据输出一行，依次输出 N 个整数，即最后的序列。

题解：

翻转操作如上述。切树操作的话，就先选择要切的区间，然后断开根的右子结点和其左子结点的联系，把要接上的节点旋转到根的右子结点出并清空其左子结点，再把切下来的子树接上去即可。

另外，由于数据量大，Splay 树又是均摊复杂度的，所以会出现退化，造成搜索深度很大，会爆堆栈，需要使用 C++ 而非 G++ 并加编译指令

```
#pragma comment(linker, "/STACK:1024000000,1024000000")
```

P0J3468 【基础】

题目大意：

给出一个数字序列，每一个数字的范围从 -10000000000 到 10000000000 ，有两种操作：

- 1、C a b c：给 $[a, b]$ 区间的每个数加上一个值 c ；
- 2、Q a b：查询 $[a, b]$ 区间所有数的和。

输入：

第一行有两个整数 N 和 M ，表示有 N 个 ($1 \leq N \leq 100000$) 整数和 M 条 ($1 \leq M \leq 100000$) 操作。

第二行有 N 个整数。接下来有 M 行，每一行有一条操作，格式如上述。

输出：

每一条 Q 操作输出一行，包含一个整数，即区间和。

题解：

需要用 long long，并且把区间最值维护改成区间和维护。依然是在 Update 和 PushDown 的时候进行维护，这个很容易改。注意要写 1LL*，否则可能出错。

HDOJ4453 【中等】

题目大意：

有一个 N 个 ($1 \leq N \leq 100000$) 数字组成的环，有 M ($1 \leq M \leq 100000$) 个操作，两个数 $k1$ 和 $k2$ ，一开始指针在队首。一共有以下几种操作：

- 1、add x ，从当前指针位置开始的顺时针 $k2$ 个数加上 x ；
- 2、reverse，翻转，从当前指针位置开始的顺时针的 $k1$ 个数；
- 3、insert x ，在当前指针位置的顺时候方向插入一个数 x ；

- 4、delete，删除当前指针所指的数；
- 5、move x，如果 x=1，指针逆时针旋转；如果 x=2，顺时针旋转；
- 6、query，查询指针所指向的数的值。

输入：

有若干组测试数据。

每一组测试数据第一行有四个整数 N、M、k1、k2，当这四个数全部为 0 的时候输入结束。

接下来有一行，有 N 个整数，表示一开始的时候从指针所在位置开始顺时针的 N 个数。

接下来有 M 行，每一行有一条操作指令，格式如上述。

输出：

每一组测试数据先输出一行 Case #测试数据编号%：其中测试数据编号从 1 开始

每一条 query 操作输出一行，即其查询的值。

题解：

1、2、3、4、6 的操作全部可以按 POJ3580 的做法来做。move 的操作，如果是逆时针旋转，那么则把队列最后一个数删掉，再插入到头部；顺时针旋转则把第一个数删掉插入到尾部即可。

HDOJ4286 【中等】

题目大意：

给出一列 N 个 ($1 \leq N \leq 500000$) 数字，以及左右两个指针的起始位置（指针是指在两个数字中间的），有 M 个 ($1 \leq M \leq 500000$) 以下几种操作：

- 1、MoveLeft L/R：将左/右指针向左移动一位，如果超过边界则不再继续移动；
- 2、MoveRight L/R：将左/右指针向右移动一位，如果超过边界则不再继续移动；
- 3、Insert L/R x：在左/右指针的右/左边插入一个数 x；
- 4、Delete L/R：删除左/右指针右/左边的一个数；
- 5、Reverse：将左右指针中间部分翻转。

输入：

第一行有一个整数 t，表示有 t 组测试数据。

每一组测试数据第一行有一个整数 N，接下来有一行有 N 个整数，表示初始序列。

然后有一行有一个整数 M，接下来有 M 行每一行有一个操作，格式如上述。

输出：

每一组测试数据输出一行，包含若干个整数，即最后的序列。

题解：

模拟两个指针的变化即可，不难。由于这题数据量太大，内存池最多需要装 100 万个节点，所以我放弃了内存回收队列（因为无论如何肯定不会超了），并且不维护区间最值信息，这样才勉强把内存压在线下。另外就是堆栈溢出需要加编译指令的问题了。

HDOJ3436 【难】

这题我也没有搞清楚，写了几次没写对，这里贴一个我参考的页面，不另写题目大意和输入输出要求了：

http://blog.csdn.net/acm_cxlove/article/details/7800619