基础线段树练习题

曾今，有个题，【ZJOI2017】线段树。现场，我爆炸了 ……

…… 于是就有了这个题 ……（大雾

算法零

对于没有询问的测试点 ……

…… 做你想做的！

期望得分 5 分。可以通过测试点 3。

算法一

直接按照题意模拟。

时间复杂度为 $O(hq \log \log A)$。

期望得分 10 分。可以通过测试点 $1, 2$。

结合算法零可以得到 15 分。

算法二

从最容易做的问题（可能并不是？）开始考虑，即，只有区间加且区间加只会被定位到一个节点上。

由于加法标记是可减的，考虑维护每个节点到根的标记之和。

当定位到了节点 x 并对它区间加了 k 后，只有节点 x 到根的链（不包括 x）、子树 x 内节点的标记和被改变了。前者全部变成了零，后者全部被加上了 k。

因此只需要一种支持链赋值、子树加，能维护每个节点的值的数据结构就行了。

使用树剖线段树，时间复杂度为 $O(q \log^2 n)$。

当树是二叉树时，LCT 可以方便的向轻孩子打标记，因此也可以使用 LCT 维护，时间复杂度为 $O(q \log n)$。

期望得分 10 分。可以通过测试点 $4, 13$。

结合算法零，算法一可以得到 25 分。

算法三

算法二需要两个限制，考虑去掉其中一个，尝试支持区间赋值操作。

这个时候标记就不可减了，直接维护标记本身。

虽然从根到节点 x 的链可能很长，但是这条链上大部分节点的标记都为 $(1,0)$。具体来说，均摊下只有 $O(\log n)$ 个节点的标记不是 $(1,0)$。（证明？考虑 LCT 的 access 操作，此处标记不为 $(1,0)$ 的节点个数小于等于 access 节点 x 时虚实边的切换次数）。

因此就可以大力下传标记了 —— 找到最浅的且标记不是 $(1,0)$ 的节点，将它的标记下传至次浅且标记不是 $(1,0)$ 的节点。将某条链链顶的标记下传至链尾只会影响到与这条链距离为 1 的节点。对这些点进行修改是个经典问题，在树剖时额外维护一个只影响链内轻孩子的标记即可。

时间复杂度为 $O(q\log^2 n)$。

我不太知道这时 LCT 的复杂度是什么 …… 根本不会分析 …… 反正肯定不会高于树剖线段树，影响不大。

期望得分 30 分。可以通过测试点 $4,5,6,13,14,15$。

结合算法零，算法一可以得到 45 分。

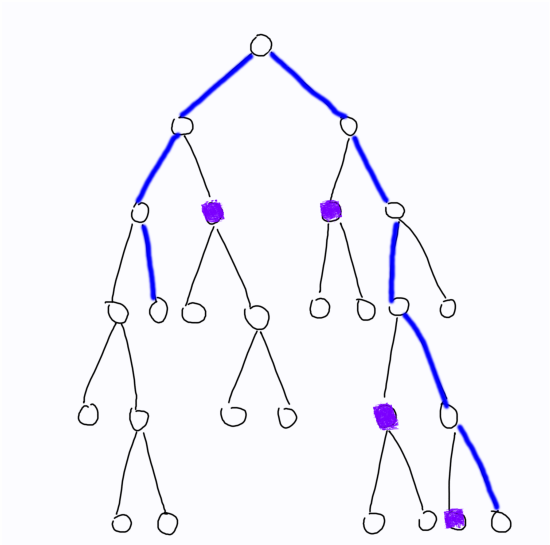
算法四

去掉另一个限制 ~

…… 使用类似解决【ZJOI2017】线段树 的技术可以解决这个问题。

区间修改时被定位到的节点、要么是左链的不在链上的右孩子，要么是右链的不在链上的左孩子。

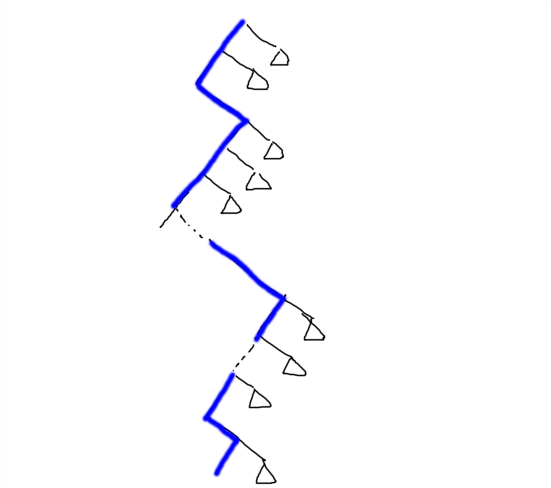
比如说对于这棵线段树。



~~画的好像有点吃藕啊~~

如果要修改区间 $[5,12]$。那么会被打上标记的是如图所示的四个紫色节点。左边的一个紫色节点是左侧那条蓝链的不在链上的右孩子，右边的三个紫色节点是右侧那条蓝链的不在链上的左孩子。

于是现在要对这些节点都打上一个标记。以左链为例，考虑其轻重链剖分后的结果：



显然，这些重链所有的是右孩子的轻孩子都需要被打上一个标记。

支持修改某条链的所有的轻孩子是可行的，同理，支持修改某条链所有的是右孩子的轻孩子也是可行的。唯一的区别在于向轻孩子下传标记的时候需要额外判断一下它是不是右孩子。

上述操作都不会影响标记下传的复杂度。修改轻孩子、修改是左孩子的轻孩子、修改是右孩子的轻孩子这三种标记是可以合并的。当然啦，实现的时候是没有修改所有轻孩子这种标记的（因为标记的影响范围不能互相包含，不然标记的顺序就无从确定了），如果要修改某条重链所有的轻孩子，只需要同时加上后两种标记就行了。

时间复杂度为 $O(q\log^2 n)$。

期望得分 60 分。可以通过测试点 $4,5,6,8,9,10,13,14,15,17,18,19$。

结合算法零，算法一可以得到 75 分。

如果只维护标记的前缀和可以得到 35 分。

算法五

区间开根！ ~~接下来基本都是口胡子~~

显然，区间开根的性质并没有被改变 …… 一个区间开 $O(\log\log A)$ 次根，最大值与最小值之差就小于等于 1 了。不过线段树的深度上天了，因此也不能暴力递归了 ……

~~这题解写的稍微有点自闭啊~~

考虑对算法四中定位到的某条重链某侧的轻孩子开根。显然，开根操作中只会递归进入那些最大值与最小值之差大于 1 的孩子，对其他孩子只需要打一个开根标记就行了（这里会有一个小问题，稍后再解决）。

当要对以节点 x 为根的子树开根时，也就是说要递归子树 x。考虑经过节点 x 的重链。显然，这条重链上有且仅有一个节点会被打上标记。这个节点是从上至下，第一个最大值减最小值小于等于 1 的节点，因此可以在维护重链的数据结构上二分出这个节点，记这个节点为 y。以节点 x 与节点 y 为端点的链上的所有节点在递归子树 x 中会被依次递归到，因此先下传这条链上的标记，再对这条链所有的轻孩子开根。要对某条重链所有的轻孩子开根，可以先对这条重链所有是左孩子的轻孩子开根，再对这条重链所有是右孩子的轻孩子开根。于是问题就解决了。

如果这样实现，对一个最大值与最小值之差为 1 的区间开根的时间复杂度等于在数据结构中定位到该区间的时间复杂度。

使用树剖线段树，时间复杂度为 $O(q \log^2 n \log \log A)$。

~~虽然我标程用的是 LCT~~

恩，还有一个小问题 …… 不能直接对一堆子树打开根标记 ……

因为开根标记是不太能下传的 …… 至少我不会把两个开根号标记加在一起。

怎么解决区间开根标记呢？当然是把它转化为加法或赋值标记咯。

对某段重链某侧的轻孩子开根时，若这些轻孩子最大值与最小值之差小于等于 1，那么就直接打一个与开根操作等价的修改标记，不然，则在维护链的数据结构中递归维护这段重链的节点的孩子（好像这里有点绕，大概意思就是这段重链跑segment tree beats）。

不过呢，当差为 1 时，不能只打一个加法标记 …… 如果其中某一棵子树内的值全是相等的，这棵子树上的标记就应该是赋值标记了。

解决方法当然也很简单 —— 再加一个标记，表示向轻孩子下传的标记是否经历过开根 …… 如果某棵子树内的值全都相等，并且向它下传的标记经历过开根，那么就算下传的是加法标记，也应该替换为等价的赋值标记。

~~显然，如果开根的时候没有进行赋值操作就没有这些事子～~

~~一道上好的大模拟题？~~

期望得分 100 分。