序列游戏 解题报告

长沙市一中 张天扬

1 试题来源

2011湖南省队集训

http://www.lydsy.com/JudgeOnline/problem.php?id=2416

2 试题大意

给一个长度为N的序列，一开始都是0。支持两种操作：

1.将区间$[L,R]$中的数依次变为$A \bmod B, 2A \bmod B, ..., (R-L+1)A \bmod B$。

2.询问区间$[L,R]$内的数的和。

一共有M次操作，对每一次操作2输出答案。每一次操作1会给出A和B。

$N \leqslant 10^9, M \leqslant 50000, 1 \leqslant A, B \leqslant 10^6$

3 算法介绍

3.1 求和式的计算

在做这道题之前，我们需要掌握一个前置技能：求下面这个式子的值

$$A \bmod B + 2A \bmod B + ... + nA \bmod B$$

注意到我们可以把它表示成：

$$\frac{A(A+1)}{2} - (\lfloor\frac{A}{B}\rfloor + \lfloor\frac{2A}{B}\rfloor + ... + \lfloor\frac{nA}{B}\rfloor)$$

因此我们只需要求出后者的值即可。分情况讨论：

3.1.1 $A \geqslant B$

令$A = kB + r(r < B)$。那么原式即：

$$\frac{kn(n+1)}{2} + (\lfloor\frac{r}{B}\rfloor + \lfloor\frac{2r}{B}\rfloor + \ldots + \lfloor\frac{nr}{B}\rfloor)$$

注意到后者可以递归计算。那么我们就把问题转化成了第二种情况$A < B$。

3.1.2 $A < B$

考虑如下数学变形（注意前提是$A < B$）：

$$\begin{aligned}
\sum_{i=1}^{n}\lfloor\frac{iA}{B}\rfloor &= \sum_{i=1}^{n}\sum_{j=1}^{\lfloor\frac{nA}{B}\rfloor}[\frac{iA}{B} \geqslant j] \\
&= \sum_{j=1}^{\lfloor\frac{nA}{B}\rfloor}\sum_{i=1}^{n}[i \geqslant \frac{jB}{A}] \\
&= \sum_{j=1}^{\lfloor\frac{nA}{B}\rfloor}(n - \sum_{i=1}^{n}[\frac{jB}{A} > i]) \\
&= n\lfloor\frac{nA}{B}\rfloor - \sum_{j=1}^{\lfloor\frac{nA}{B}\rfloor}(\lfloor\frac{jB}{A}\rfloor - [jB \bmod A = 0]) \\
&= n\lfloor\frac{nA}{B}\rfloor + \sum_{i=1}^{n}[iA \bmod B = 0] - \sum_{j=1}^{\lfloor\frac{nA}{B}\rfloor}\lfloor\frac{jB}{A}\rfloor
\end{aligned}$$

上式中第一项很好计算，第三项实际上是$n_1 = \lfloor\frac{nA}{B}\rfloor, B_1 = A, A_1 = B$的递归计算。而第二项我们也容易知道它的值为$\lfloor\frac{\frac{n}{B}}{gcd(A,B)}\rfloor$。于是我们就将$A < B$的情况再次转换成了$A \geqslant B$的情况。

注意到，在上述两种情况间不断迭代时，每迭代两次A, B的值会减小一半（根据欧几里得算法可以得出）。那么复杂度为$O(\log B)$。

3.2 使用线段树维护

我们回到原问题。考虑使用一棵线段树来维护整个序列，每次区间覆盖操作时，我们需要修改线段树上的$O(\log n)$个节点，修改每个节点时需要求2次上

面所说的和，相减得到这个节点的区间和。因此算法复杂度为$O(M\log n\log B)$。

注意由于n很大，线段树需要动态开点。也可以考虑离线，将所有要用到的位置离散化，那么复杂度就是$O(M\log M\log B)$。

3.3 使用平衡树维护

我们不妨考虑使用平衡树来维护整个序列。平衡树中的每个节点代表序列中某一次被覆盖的一个区间。那么当我们覆盖一个区间的时候：

3.3.1 新区间被一个原区间包含

我们把包含它的原区间分为三部分：新区间的左边，新区间的右边以及新区间本身。我们直接计算这三部分的区间和后，删去原区间在平衡树中的节点，把这三个新节点插入进平衡树。

注意到，由于我们需要计算区间和的次数是$O(1)$的，那么这部分的复杂度是$O(\log B+\log M)$。

3.3.2 新区间未被一个原区间包含

有的原有区间被新区间完全包含，那么这样的原有区间可以直接从平衡树中删去。

有的原有区间和新区间有交集，显然，这样的区间最多只有两个（左右各一个），那么我们直接修改原有区间的信息即可。

之后，我们把新区间插入到平衡树中即可。

这种情况下，我们也只需要计算3次区间和，复杂度也是$O(\log B+\log M)$。

因此，我们使用平衡树维护的总复杂度为$O(M(\log M+\log B)$，比使用线段树维护的复杂度要优秀。在实际测试中，由于平衡树这种数据结构本身带来的巨大常数，比使用线段树大概只快3倍左右。

3.4 总结

考虑使用平衡树维护区间问题，限制性比较大，只有在覆盖一个区间的复杂度高于$O(1)$的时候，才能出现比线段树优秀的复杂度。而且代码也比较繁琐易错。但是在一些特殊情况下，也不失为一种降低时间复杂度的好办法。