《strakf》解题报告

佛山石中李子豪

1 试题来源

2016国家集训队互测

2 试题大意

给了一个长度为N的仅由小写字母组成的初始字符串S.
之后有M次操作，操作分为三类：
1 A:将原串S的所有等于A的子串权值+1；
2 l r:将原串S的所有等于$S[l,r]$的子串权值+1；
3 a b:询问原串S所有子串$S[l,r]$满足$a \le l \le r \le b$的权值和。

3 数据范围

数据点编号	N, M	操作类型	数据特点
1	5	2,3	无
2	10	2,3	无
3	100	2,3	无
4	$2*10^5$	1,2	无
5	$2*10^4$	1,3	所有的3操作a=1
6	$2*10^4$	1,3	所有的3操作b=N
7-8	$2*10^4$	1,3	所有的1操作的字符串长度为不下降序列
9-10	$6*10^3$	2,3	无
11-12	$2*10^4$	2,3	所有的3操作b=N
13-15	$2*10^4$	2,3	所有的2操作的字符串长度为不下降序列

16-20	$2*10^4$	2,3	无

输入文件大小不超过0.3M。

4 试题考点

字符串、数据结构、分块

5 算法介绍

5.1 第1-3个数据点

采取无脑暴力的方法。
对于每一个修改操作，则把所有符合的子串的权值+1.
对于每一个询问操作，则直接询问范围内的所有子串的权值和。
时间复杂度为$O(n^3)$,预计得分15分。

5.2 第4个数据点

注意到这个数据点并没有操作3，因此直接结束即可。
算上前面的15分，预计得分20分。

5.3 第5-8个数据点

这一类数据点的特点在于只有操作1的修改，而没有操作2的修改。

5.3.1 数据特点

由于输入保证不超过0.3M,可以得到插入总长度不超过$3*10^5$。
因此，我们可以得到一个结论：插入字符串的长度最多只有$O(\sqrt{N})$种。
证明：
  假设长度恰好为1到$\sqrt{N}$各一个，那么总长度已达到$O(N)$级别。
  证毕。

5.3.2 解决方法

因此，我们对于询问操作，可以对于各个不同长度的询问分开处理。

那么，问题变成要维护各个不同长度的左端点在某一区间范围的方案数。

而对于插入操作，我们可以得出：插入操作所能影响的左端点的后缀排名处于某一个区间范围内。

因此，我们可以对于每一个左端点记录一个二元组(a,b)表示左端点位置在a，对应后缀排名为b。

然后插入则是对于所有$b \in [l, r]$的二元组进行权值+1.询问则是对于所有$a \in [l, r]$的二元组询问权值和。

我们可以采取分块来解决这个问题。

假设我们按照位置序列来分块，块长为L.

那么我们可以对于每一块记录块内权值和以及按照b排序的块内端点排名从而记录对于每一个给定的r得到的最大的k满足$b_k \le r$.

然后，再对前i块记录权值和sum_i.

对于修改操作，我们直接求出对于每一块的贡献以及对于前i块的贡献值，修改对应的块内权值和以及sum_i.并且对于每一块记录一个修改标记。

对于询问操作，则先利用求好的sum_i求出整块的权值和，然后对于非整块部分，则直接打算标记，暴力询问即可。

分析复杂度，修改操作复杂度为$O(\frac{N}{L})$,询问操作复杂度为$O(L\sqrt{N})$. 当$L = N^{0.25}$时，有最小复杂度为$O(N^{0.75})$.

因此，整个算法复杂度为$O(N^{1.75})$.预计得分20分。

算上上面其余部分的分数，预计得分40分。

5.4 第9-10个数据点

对于这一类数据，同样的，我们可以先构造一个后缀数组。

然后，我们对于每一个位置维护一个结构，维护以该点为左端点的所有修改长度。

对于修改操作，我们可以通过二分求出包含这一子串的后缀排名区间。我们可以对于排名区间里面对应的每个位置，在其对应的结构里面插入当前的修改长度。

然后，对于询问操作，则对操作区间内的所有点分别询问其结构里面长度不超过某一个值的修改个数。

因此，我们需要一个可以维护动态插入以及询问比某个值小的值的个数，可以使用线段树或者平衡树，也可以采用pb_ds库里面的黑科技。

时间复杂度为$O(NMlog_2N)$.预计得分10分。

算上上面其余部分的分数，预计得分50分。

5.5 第11-12个数据点

这一类数据的特点在于右端点固定为N，即不存在右端点越界的情况。因此，我们不需要分开不同长度来处理。

因此，我们可以采取类似第5-8个数据点的解决方法：

对于每一个左端点记录一个二元组(a,b)表示左端点位置在a，对应后缀排名为b。

然后插入则是对于所有$b \in [l, r]$的二元组进行权值+1.询问则是对于所有$a \in [l, r]$的二元组询问权值和。

我们上面原本使用的是分块，但其实这里也是可以采用KD-tree来解决的。使用KD-tree可以减少一定的代码量以及运行速度上也会有一定的提升。

可以证明，对于坐标两两不相同的情况，KD-tree可以保证最坏$O(\sqrt{N})$的复杂度完成二维区间内的问题。

这一点，我们可以分两步来证明：

5.5.1 对于其中一维区间为全集的情况

首先，比较容易可以知道另一维区间为任意区间[L,r]实际上可以等价于[1,r]区间的解决。

这个可以简单证明得到:假设线段树中mid为使得L,r处于两个不同区间的节点，那么下一步问题变成了等价于[L,n]以及[1,r]的问题。

然后，对于[1,r]的情况，如果当前是以全集区间的那维分两边，那么显然两边节点都要遍历，而如果是以[1,r]区间的那维的话，那么必然有一边要么完全不覆盖，要么完全覆盖，都可以一步解决，因此，实际需要往下的只有一个节点。

然后，总层数为log_2N层，然后只有$\frac{log_2N}{2}$层需要遍历两边，因此复杂度为$O(2^{\frac{log_2N}{2}}) = O(\sqrt{N})$.

5.5.2 一般情况

首先，同样容易证明其中一维的任意区间[L,r]可以等价于[1,r]。

然后当当前是以这一维分两边的话，那么必然有一边这一维要么完全不覆盖，要么完全覆盖。最坏情况是完全覆盖那一种，那么这一部分我们已经证明是$O(\sqrt{N})$级别的了。而另一边，则是等价的问题。

因此，我们有$F(N) = F(N/2) + O(\sqrt{N})$,根据等比数列我们可以得到$F(N) = O(\sqrt{N})$.

证毕。

于是，我们可以通过$O(M\sqrt{N})$的复杂度解决。预计得分10分。

算上上面其余部分的分数，预计得分60分。

5.6 第13-15个数据点

这一类数据点，保证了插入长度不下降。这个提示我们可以往长度方向想。

考虑当前询问区间为[L,r]，我们设[1,L-1]为A区间，[L,r]为B区间，[r+1,N]为C区间。然后设(x,y)意义为左端点在x区间，右端点在y区间的权值和。例如(A,B)表示左端点在A区间，右端点在B区间。我们求解的是(B,B)的值。

那么，对于[1,r]区间的询问，我们得出了(A,A)+(A,B)+(B,B)。那么，如果我们有办法求解(A,A)+(A,B)就可以解决了。因为我们已经知道其中一个端点在边界的解法了。

然后我们可以求[1,N]区间的询问，从而得出(A,A)+(A,B)+(A,C)+(B,B)+(B,C)+(C,C).

然后又由(B,B)+(B,C)+(C,C)为[L,N]区间的询问，可以得到。

因此，现在唯一的问题就是(A,C)问题的求解了。但这一类问题似乎没有什么好的解决方法。

这时，我们回过来看长度的提示。然后我们可以得到一个突破点：如果保证长度不超过$r - l + 1$的话，那么不存在(A,C)这个问题。

因此，我们可以根据长度来维护一个可持久化KD-tree，维护只包含长度不超过某个值的答案。

那么，对于询问操作，我们只需要先二分出对应最大长度对应的KD-tree，然后在这棵KD-tree进行询问即可。

时间复杂度为$O(M\sqrt(N))$,预计得分15分。

算上上面其余部分的分数，预计得分75分。

5.7 第16-20个数据点

最后这一类数据点，实际上只需要对第13-15个数据点的解决方法进行少量修改即可。

我们观察到这一题允许离线，并且操作独立，因此，我们只需要在外面套上一个cdq分治，然后就能保证插入长度单调，从而就能直接利用第13-15个数据点的解决方法解决了。

由于多套了个cdq分治，时间复杂度为$O(M\sqrt(N)log_2M)$.

进行少量修改，可以通过全部数据。预计得分100分。

6 总结

这道题主要考察对题目性质的挖掘以及转化为简单问题的解决方法。

总的来说，难度与NOI第二题接近。预计一半左右选手能拿到70分以上，所有选手均能拿到40分或以上。