cx 觉得

良心签到大水题。

从左到右把S_1的所有字符压入栈中。加入一个字符之后，如果栈顶$|S_2|$个字符形成的串和S_2相等则把这$|S_2|$个字符弹栈。结束之后栈中的所有字符连成的串即为答案。

$O(|S_1||S_2|)$。

他的水平

线段树优化 DP 经典题。

暴力的 DP 式子：

$$\mathrm{f[i]} = \max_{0\le j<i}\{g[j]-(s_i-s_j)+cst(j+1,i)\}$$

其中$\mathrm{f[i]}$为就所有输入给出的，右端点不超过i的区间产生的最大收益（且第i块地必须不荒）。s为v的前缀和数组，$\mathrm{cst(l,r)}$为包含于$\mathrm{[l,r]}$的所有区间的收益之和，g为f的前缀最大值。

直接做是$\mathrm{O(n^2)}$的，我们需要考虑优化。

首先，上面这个式子的$-\mathrm{s_i}$可以移到max外面，我们需要在i不断递增的情况下，实时维护$\mathrm{g[j]+s_j}+cst(j+1,i)$的最大值。

考虑使用线段树，第j个位置的值即为$\mathrm{g[j]+s_j}+cst(j+1,i)$。

$\mathrm{g[j]+s_j}$一旦确定就不会再改变，故我们的重点在$\mathrm{cst(j+1,i)}$。

考虑i加一的影响：显然，$\mathrm{cst(j+1,i)}$只比$\mathrm{cst(j+1,i-1)}$多了右端点为i，左端点大于j的所有区间收益之和。故需要计算$\mathrm{f[i]}$时，先枚举所有右端点为i的区间，对于一个这样的区间$\mathrm{[l,i]}$，对线段树的区间$\mathrm{[0,l)}$整体加上这个区间的收益即可。然后$\mathrm{f[i]}$就是区间$\mathrm{[0,i-1]}$的最大值减去$\mathrm{s_i}$。$\mathrm{f[i]}$的值计算好之后，将线段树的第i个位置加上$\mathrm{g[i]+s_i}$即可。

$\mathrm{O((N+M)}\log N)$。

比 zzq

定义$\mathrm{query}(N, l, r, x, y)$为询问$F(A = \{1,2, \dots, N\})$区间$[l, r]$内所有值在$[x, y]$之间的数之和。

首先，如果$l = 1, r = N$则直接返回$[1, N] \cap [x, y]$内所有数的和。

否则由于$F(A) = F(\mathrm{odd}) + F(\mathrm{even})$，可以把A递归到odd和even。

考虑和线段树一样递归处理。易得$|\mathrm{odd}| = \lceil \frac{N}{2} \rceil$，$|\mathrm{even}| = \lfloor \frac{N}{2} \rfloor$。

（1） $r \le |\mathrm{odd}|$：往odd递归。

（2） $l > |\mathrm{odd}|$：往even递归。

（3） $l \le |\mathrm{odd}| < r$：往两边递归。

我们会发现一个问题：递归下去的子问题变成了对$F(\{1,3,5, \dots\})$或$F(\{2,4,6, \dots\})$这样的序列进行查询，不满足query的定义。

但我们注意到，$F(\{2,4, \dots ,2|\mathrm{even}|\})$区间$[l, r]$内所有值在$[x, y]$之间的数之和等于$F(\{1,2, \dots, |\mathrm{even}|\})$ 区间$[l, r]$内所有值在$[\lceil \frac{x}{2} \rceil, \lfloor \frac{y}{2} \rfloor]$之间的数之和的2倍，故对于even递归下去的时候把x和y进行这样的处理即可。

对于$F(\{1,3,5, \dots ,2|\mathrm{odd}| - 1\})$，它的区间$[l, r]$内所有值在$[x, y]$之间的数之和等于$F(\{1,2, \dots, |\mathrm{odd}|\})$ 区间$[l, r]$内所有值在$[\left\lfloor \frac{x}{2} \right\rfloor + 1, \lceil \frac{y}{2} \rceil]$之间的数的2倍减一之和，也就是这些数之和的2倍之和再减去这些数的个数。故$\mathrm{query}(N, l, r, x, y)$实际上需要返回一个二元组，表示$F(\{1,2, \dots, N\})$的区间$[l, r]$内所有值在$[x, y]$之间的数的个数与和。

$O(M \log N)$。

高

简单倍增。

问题可以转化成枚举一个u再枚举一个v，将$\mathrm{Dist}(u,\mathrm{lca}(u,v))$计入答案。

考虑把枚举v改成枚举$w=\mathrm{lca}(u,v)$，那么这样的w对答案的贡献就是$\mathrm{Dist}(u,w)\times(\mathrm{size}[w]-\mathrm{size}[x])$。其中x为u到w的路径上倒数第二个点。

不过由于w的枚举范围是u的严格祖先，在最坏情况下这样的复杂度还是$O(n^2)$的。

定义数组：

$\mathrm{fa}[u][i]$：u的2^{i}级祖先。

$\mathrm{p}[u][i]$：u的2^i-1级祖先。

$\mathrm{f}[u][i]$：$\sum_{w\ is\ an\ ancestor\ of\ u,w\neq u,d(u,w)<2^i}(size[w]-size[lst(u,w)])\times Dist(u,w)$

（$\mathrm{lst}(u,v)$为u到v的路径上倒数第二个点，$\mathrm{d}(u,v)$为u到v的距离）

换句话说，$\mathrm{f}[u][i]$表示u已经确定，对于所有满足$\mathrm{d}(u,\mathrm{lca}(u,v))<2^{\mathrm{i}}$的v，它们的$\mathrm{Dist}(u,\mathrm{lca}(u,v))$之和。

fa和p的转移比较简单：$\mathrm{fa}[u][i+1]=\mathrm{fa}\big[\mathrm{fa}[u][i]\big][i]$，$\mathrm{p}[u][i+1]=\mathrm{p}\big[\mathrm{fa}[u][i]\big][i]$。

边界$\mathrm{fa}[u][0]=\mathrm{father}[u]$，$\mathrm{p}[u][0]=u$。

而对于$\mathrm{f}[u][i+1]$，分两种情况考虑：

（1） $\mathrm{d}(u,w)<2^{\mathrm{i}}$，这一部分的贡献就是$\mathrm{f}[u][i]$。

（2） $2^{\mathrm{i}}\leq d(u,w)<2^{i+1}$，这一部分的贡献，除去第i位的影响之外为$\mathrm{f}\big[\mathrm{fa}[u][i]\big][i]$。

现在考虑第i位的影响。易得这就是所有满足$2^{\mathrm{i}}\leq d(u,w)<2^{i+1}$的$\mathrm{size}[w]-\mathrm{size}[\mathrm{lst}(u,w)]$之和。即$\mathrm{size}\big[\mathrm{p}[u][i+1]\big]-\mathrm{size}[\mathrm{p}[u][i]]$。

故我们得出了f的转移：

$$\mathrm{f}[u][i+1]=\mathrm{f}[u][i]+\mathrm{f}\big[\mathrm{fa}[u][i]\big][i]+\Big(\mathrm{size}\big[\mathrm{p}[u][i+1]\big]-\mathrm{size}\big[\mathrm{p}[u][i]\big]\Big)\times[\mathrm{fa}[u][i]\neq 0]$$

最后答案即为$\sum_{u=1}^{n}f[u][[\log n]]$。

$O(\mathrm{n}\log n)$。