改

首先显然如果一个串合法，那么在其中插入若干字符后依然是合法的。

所以我们可以设 $dp_{i,j}$ 表示对于选定的串 A 的一个左端点，串长最短是多少，能使得其存在一个子序列能由串 B 的前 i 个字符插入队列得到。

转移就是枚举 B 的下一个字母是插入开头还是结尾，复杂度 $O(n^2)$。

祭

我们发现方案数即为选择两个矩形，要求存在一条水平或竖直直线能把它们切开。

于是我们求对于一个分界线，有多少合法矩形完全在分界线左边即可。

我们进行扫描线，维护有多少合法矩形右端贴在扫描线上。设 h_i 表示第 i 行从扫描线往左延伸多少会碰到障碍。设 l_i 与 r_i 表示左/右第一个 j 使得 $h_j > h_i$ 。我们每次把扫描线往右移动，则所有 h_i 加一，每碰到一个障碍，则把一个 h_i 清零，并且把其左侧的 r_j 变为 $\min(r_j, i)$，右侧的 l_j 变为 $\max(l_j, i)$。Segment Tree Beats维护即可。复杂度 $O(n\log^2 n)$。

瞬

有如下结论：

1.我们一定是按照某DFS序遍历的。

2.由1可得，分身一定放在一条路径上。

3.如果一个点放置了分身，在遍历其某子树的时候又撤走了，那么在这棵子树遍历完之前，我们不会去使用这个分身。

4.由3可知，我们如果要撤走一个分身，一定是撤走离自己最近的。

我们可以设计一个DP，设 $f_{i,j,k}$ 表示你还有 k 个分身可用，你现在在 i 号点，i 号点必有一个分身， 你上一个分身在 j。转移我们就枚举是否撤走 i 位置的分身且放在子树根处，复杂度为 $O(n^2k)$。

我们发现当 $k > \log_2 n$ 时，答案都是所有边权和。我们可以对原树树剖，我们最后遍历一个节点的重儿子，并且把分身移下去。轻儿子我们就另开一个分身。这样分身数不会超过一个点到根的最大轻边数，即 $\log_2 n$。于是复杂度可化为 $O(n^2\log n)$。