题意比较简单明了不说了。

1. 先用给定字符串构建后缀自动机，建出parent链，对主节点按顺序赋予权值1到n。

根据后缀自动机性质可知：parent链子取反可得到后缀树；且对于任意节点，设root沿着主链到达这个节点代表字符串s,设沿着的这个节点的p指针走的路径上的主节点构成集合ns，从根节点到达ns里的节点全部路径代表的字符串构成了s的全部后缀。

1. 简单的沿着后缀自动机的dfa走，走到节点为node, 对于询问l,r,k，设[l,r]代表的字串是s[l,r]，则必然以node开始沿着后缀树(parent的反方向)走的每个主节点的权值是s[l,r]的一次出现的末尾位置编号。查询第k次出现就是找到node的子树dfs序的主节点权值第k大
2. 用对后缀树进行深度优先且后序遍历，把主节点的权值建成连续数组，带可持久化权值线段树模板，求区间第k大即可。
3. 到这里还没完，找到第2步中的节点node需要花费o(n)的时间，使得Q次询问的总复杂度到达Q\*n\*logn明显超时。

所以这步的问题是：给定l,r,k如何通过s[l,r]高效找到对应的后缀自动机节点，这要用暴力走dfa以外的方法(因为暴力走dfa复杂度on)

应该改为用树上倍增的算法寻找,

node.binp[i][j]代表第编号为i的后缀树的节点向上走2^j个节点的编号。预处理出倍增数组binp,利用它寻找出距离根节点r-l+1深度的节点，它就是node.如此做时间复杂度logn,总查找时间复杂度是logn\*logn\*Q

本题的总时间复杂度是

logn\*logn\*Q //总查找时间复杂度

+n\*logn//预处理倍增数组

+n//主席树建树

+n//建后缀数组，顺便建后缀树。

舍去低阶无穷大，时间复杂度O(logn\*logn\*Q)

本题考查了：后缀自动机，后缀树，可持久化权值线段树，树链剖分思想，树上倍增思想。