fraction

subtask1

考虑分治，每次找一根最长的分数线，它会最后算，答案是左边乘右边的逆，递归。

复杂度 $O(nq)$

subtask2

考虑优化 subtask1 的做法，把分治结构建成树状结构，查询时若询问区间包含当前区间直接返回，类似的在线段树上查询即可。

复杂度 $O(q \times dep)$ ，随机数据下 dep 期望为 $O(\log n)$

这棵树实际上是棵广义线段树。

subtask3,4

做法很多，这里随便挑了一种。

考虑广义线段树上一个区间 $[L, R]$，它的最大值为 p

考虑如果我的询问 X, Y 满足：$L \le X \le p < Y \le R$ ，如果我们能预处理出 $[X, p]$ 的答案和 $[p+1, Y]$ 的答案的话，就可以计算答案。

于是我们有了一个很清晰的思路：对于每个区间预处理出它的前缀和后缀的答案。

一个区间的后缀有两种情况：

1. 是右区间的后缀。
2. 是左区间的后缀乘上整个右区间的答案。

考虑对每个结点开一棵线段树来记录后缀的答案。

结点的后缀=右儿子的后缀+左儿子的后缀/右区间的答案。

这实际上是一个线段树合并，对前缀也可以用类似的做法。

复杂度 $O((n+q)\log n)$。

subtask5,6

继续使用 subtask2 的做法。

有个一个广义线段树上的结论：考虑在广义线段树上查询区间 $[l,r]$ ，考虑 $[l-1,l-1]$ 对应的点 x 和 $[r+1,r+1]$ 对应的点 y，记 l 为 x,y 的最近公共祖先。

若 x 到 l 链上的某个点有不在该链上的右儿子，则该右儿子会被定位到；若 y 到 l 链上的某个点有不在该链上的左儿子，则该左儿子会被定位到，并且不难发现以上节点恰好就是被定位到的所有节点。

预处理每个点到根上的点的兄弟的值乘积，乘积的逆，一个乘一个除的乘积，一个除一个乘的乘积即可。

对于每次询问可以简单分类讨论得到答案。

使用 st 表实现预处理和查询，可以做到时间 $O(n\log n+q)$ 空间 $O(n\log n)$ ，只能通过 subtask5。

使用 $O(n)-O(1)\ rmq$，可以做到时间 $O(n+q)$，空间 $O(n)$，可以轻松通过本题。