YJJ’s Stack题解

一种可行解法：

先考虑没有 `pop` 的情况，因为 v 值最多五个，我们完全可以针对每一个的 v 值维护一个栈，每次查询的时候将每个栈的栈顶元素的插入时间 t 进行比较， t 更大的那个就是总栈的栈顶。

所以只用考虑维护一个栈的时候，我们把 t 离散化。当 `query T` 时，我们需要找到一段区间最短的区间 $[x,T]$ ，使得 $[x,T]$ 这段区间里 `push` 和 `delete` 操作抵消后，只剩一个 `push` 的元素，这个元素的插入时间就是 x。

找到 x 值的方式多种多样，其中一种考虑权值线段树， `push` 操作在线段树 t 上+1， `delete` 操作-1，所以只用找到最短的区间，使得 $sum[x,T]=1$ 即可。具体可以用区间修改，令 val[i] 表示 $[i,N]$,即以i为后缀的区间的和，所以 $val[x]-val[T]$ 即可表示 $[x,T-1]$ 区间的和，要找到栈顶，等价于找到最小的 $x<T$,使得 $val[x] > val[T]$ 即可，改一改线段树的查找写法，优先处理右区间即可。

我们把没有 `pop` 时的 `query` 操作命名为 `query0` ，再考虑有 `pop` 时，因为 `pop` 数量只有 5，每次 `query` 时，需把 `pop` 替换成相应等价的 `delete v` ，所以就需要在 `pop` 的地方进行子操作 `query0` ,获得 `pop` 对应删除的是哪个 v，替换成 `delete` 后再在 T 处进行一个 `query0` 操作即可。

总复杂 $O(c(k+1)n\log(n))$, c 是 v 的值域，k 是 `pop` 的数量。