Problem 2.1 [Integer Retroactivity].

Describe and analyze a fully retroactive data structure for storing an integer, initially 0. Your data structure should support \textsc{insert}(t, update) and \textsc{delete}(t, update), where $t$ denotes the time of the operation and $update$ is one of the following update operations:

(a) \textsc{add}(x): add $x$ to the stored integer.

(b) \textsc{multiply}(x): multiply the stored integer by $x$.

(c) \textsc{set}(x): set the stored integer to $x$.

You should be able to query the value of the integer at a time $t$.

Your data structure should support all operations (retroactive updates and queries) in $O(\log m)$ time per operation, where $m$ denotes the number of update operations (\textsc{add}, \textsc{multiply}, and \textsc{set}) that have been added to and not deleted from the data structure. The space usage should be $O(m)$. 
