

Problem Set 2

Due: Wednesday, September 20, 2017 at noon

Problem 2.1 [Integer Retroactivity].

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

- (a) $\text{ADD}(x)$: add x to the stored integer.
- (b) $\text{MULTIPLY}(x)$: multiply the stored integer by x .
- (c) $\text{SET}(x)$: 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 (ADD , MULTIPLY , and SET) that have been added to and not deleted from the data structure. The space usage should be $O(m)$.