6.851: Advanced Data Structures, Spring 2021

Prof. Erik Demaine, Josh Brunner, Dylan Hendrickson, Yevhenii Diomidov

Problem Set 2

Due: Thursday, March 4, 2021

Problem 2.1 [Functional Queues]. Design a functional first-in first-out queue data structure. It should support the following two operations in $O(1)$ worst-case time per operation:

(a) **insert-last**(Q, x): append x as the last element in the queue Q.

(b) **delete-first**(Q): remove the first element in the queue Q, and return it.

Each operation should also return the new version Q' of the data structure resulting from the update. Remember that, for a data structure to be functional, no operation actually modifies the data structure, so every previous version is always accessible. This means that your data structure should automatically be fully persistent.

Hint: Start by storing two stacks, one for the front of the queue and one for the rear of the queue. You can get an amortized solution by periodically reversing the rear stack and appending it to the front stack. Find a de-amortized version of this strategy where you do only a constant amount of the reversal or append during each operation.