York University EECS 4101/5101 Eric Ruppert September 5, 2024

Homework Assignment #1
Due: September 13, 2024 at 5:00 p.m.

1. We wish to implement a Block ADT, which stores a sequence $x_0, x_1, \ldots, x_{K-1}$ of fixed length K and supports the following operations.

- Index(i) returns x_i.
- Insert(i, x) inserts element x in the ith position of the sequence, and removes and returns the last element. This operation changes the sequence from $x_0, x_1 \ldots, x_{K-1}$ to $x_0, x_1, \ldots, x_{i-1}, x, x_i, x_{i+1}, \ldots, x_{K-2}$ and returns x_{K-1}.
- ShiftRight(x) is equivalent to Insert($0, x$).

Give a *simple* implementation of this ADT so that Index and ShiftRight each run in $O(1)$ time and Insert runs in $O(K)$ time. Explain how your data structure represents the sequence in memory. You can either give a brief (but precise) English description of how each operation is implemented or pseudocode.

2. Now, we wish to implement a Sequence, which stores a (variable-length) sequence of elements $x_0, x_1, x_2, \ldots, x_{n-1}$ and supports the following operations.

- Index(i) returns x_i.
- Insert(i, x) inserts element x in the ith position of the sequence, so the sequence changes from $x_0, x_1 \ldots, x_{n-1}$ to $x_0, x_1, \ldots, x_{i-1}, x, x_i, x_{i+1}, \ldots, x_{n-1}$.

Suppose we know in advance that there is a value N such that n (the number of elements in the sequence) starts at N and never exceeds $2N$. Our goal is to design a data structure such that the worst-case time for Index is $O(1)$ and the worst-case time for Insert is *sublinear* (i.e., $o(n)$).

Let K be a fixed integer, which you will choose later on. We divide the sequence into $\lceil n/K \rceil$ Blocks $B_0, B_1, \ldots, B_{\lceil n/K \rceil - 1}$, each of size K. (The last Block might not be full, so we can pad it with null values.) Each Block will be represented using your answer to Question 1. You will also use an array *blocks*$[0..\lceil 2N/K \rceil]$, where *blocks*$[i]$ stores a pointer to Block B_i. Thus, you can apply the three Block operations defined in Question 1 to *blocks*$[i]$.

The first $\lceil N/K \rceil$ entries of *blocks* are initialized with pointers to Blocks containing the entries of the sequence. All entries of Blocks beyond the end of the sequence are initially null. (In particular, for $i \geq \lceil N/K \rceil$, *blocks*$[i]$ points to a Block with K null entries.)

(a) Give pseudocode for the two operations on the Sequence object.

(b) What is the worst-case running time of each operation? State your answer as a function of N and K using Θ notation.

(c) If you know N, how would you choose K to make the worst-case running time of Insert as small as possible?

Hint: If your solutions to this assignment is more than a page long, it is probably more complicated than it needs to be.