York University EECS 4101/5101 Eric Ruppert January 10, 2024

Homework Assignment #1
Due: January 17, 2024 at 5:00 p.m.

1. Consider an *ordered stack* that stores a sequence of integers and supports the following two operations.

- POP removes and returns the last item of the sequence.
- ORDEREDPUSH(x) removes *all* integers that are larger than x from the sequence, and then adds x to the end of the sequence.

Assume the stack is initially empty.

[1] **(a)** Show the sequence stored in the ordered stack after performing the following sequence of operations.
ORDEREDPUSH(1)
ORDEREDPUSH(4)
ORDEREDPUSH(6)
ORDEREDPUSH(3)
ORDEREDPUSH(5)
POP

[3] **(b)** State an interesting invariant about the order of elements in the sequence. Explain briefly why it is an invariant. (That is, explain why every operation maintains the truth of the invariant.)

[1] **(c)** Now, suppose we want to implement the ordered stack as a singly-linked list, with the goal of ensuring that operations have good amortized time. Think about which direction the pointers of the singly-linked list should go. Draw a picture of the stack after the sequence of operations of part (a), including all pointers.

[3] **(d)** First, describe in one sentence, the main idea of how to perform an ORDEREDPUSH efficiently. Then, give pseudocode for ORDEREDPUSH and POP. Comment your pseudocode.

[1] **(e)** What is the worst-case time for an individual ORDEREDPUSH or POP operation? State your answer using Θ notation in terms of n, the number of elements in the stack.

[4] **(f)** Give good bounds on the amortized time per operation and prove your answer is correct.