Data Structures and Algorithms Spring 2023 — Problem Sets

by Nikolai Kudasov

January 25, 2023

Week 2. Problem set

1. In [Cormen, Section 16.1], a total amortised time complexity is computed for a sequence of n INCREMENT[1] operations starting from an initially zero counter. Compute the total amortised time complexity for a sequence of n INCREMENT operations starting from a non-zero counter with value k. No justification is required, provide your answer using big-Oh notation.

2. In [Cormen, Section 16.1], a stack with an extra operation MULTIPOP is discussed. Provide an example of a sequence of PUSH, POP, and MULTIPOP operations on an initially empty stack, such that

   - the actual total cost of the sequence is 5,
   - the sequence contains one POP, and one MULTIPOP, and 3 PUSH operations, in some order,
   - MULTIPOP(k) must be used with $k \geq 2$.

   No justification is required for this exercise.

3. Consider `StackQueue`, an implementation of the Queue ADT using a pair of stacks: a *front stack* and a *rear stack*:

   - A queue is empty when both stacks are empty.
   - To perform `offer(e)`, we `push(e)` into the rear stack.
   - To perform `poll()`, we `pop()` from the front stack if it is not empty. If the front stack is empty, we repeatedly `pop()` elements from the rear stack and `push` them onto the front stack, until the rear stack is empty. Finally, we `pop()` from the front stack, since it is no longer empty.

   Perform amortised time complexity analysis for a sequence of `offer(e)` and `poll()` operations performed on an initially empty `StackQueue`. You **must** apply either the accounting method or the potential method.

   Assume that the execution cost (time) of `push(e)`, `pop()`, `isEmpty()` for the underlying stack implementation is 1.

References

[Cormen] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms, Fourth Edition.* The MIT Press 2022

[1] when using LaTeX, write `\textsc{Increment}` to typeset INCREMENT in small capital letters