# DM582 Exercises - Sheet 7

Mads Anker Nielsen    Tobias Samsøe Sørensen

March 5, 2025

This document contains exercises from the course DM582 (spring 2025). Most exercises are from the book *Introduction to Algorithms, 4th edition* by Cormen, Leiserson, Rivest, and Stein (CLRS), the book *Algorithm Design, 1st edition* by J. Kleinberg and E. Tardos (KT), and the book *Discrete Mathematics and its Applications, 8th edition* by K. Rosen.

References to CLRS refer to the book *Introduction to Algorithms, 4th edition* by Cormen, Leiserson, Rivest, and Stein.

References to KT refer to the book *Algorithm Design, 1st edition* by J. Kleinberg and E. Tardos.

References to Rosen refer to the book *Discrete Mathematics and its Applications, 8th edition* by K. Rosen.

This document will inevitably contain mistakes. If you find some, please report them to your TA so that we can correct them.

# Sheet 6

## Exercise 16.1-2

**Exercise**

Show that if a `DECREMENT` operation is included in the bit counter example, operations can cost as much as $\Theta(nk)$ time.

## Exercise 16.3-2

**Exercise**

Redo Exercise 16.1-3 using a potential method of analysis.

Exercise 16.1-3: *Use aggregate analysis to determine the amortized cost per operation for a sequence of operations on a data structure in which the $i$-th operation costs $i$ if is an exact power of $2$, and $1$ otherwise.*

## Exercise 16.3-3

**Exercise**

Consider an ordinary binary min-heap data structure supporting the instructions `INSERT` and `EXTRACT-MIN` that, when there are items in the heap, implements each operation in $O(\log n)$ worst-case time. Give a potential function such that the amortized cost of `INSERT` is $O(\log n)$ and the amortized cost of `EXTRACT-MIN` is $O(1)$, and show that your potential function yields these amortized time bounds. Note that in the analysis, $n$ is the number of items currently in the heap, and you do not know a bound on the maximum number of items that can ever be stored in the heap.

## Exercise 16.3-5

**Exercise**

Show how to implement a queue with two ordinary stacks (Exercise 10.1-7) so that the amortized cost of each `ENQUEUE` and each `DEQUEUE` operation is $O(1)$.

## Exercise 16.4-3

**Exercise**

Discuss how to use the accounting method to analyze both the insertion and deletion operations, assuming that the table doubles in size when its load factor exceeds 1 and the table halves in size when its load factor goes below 1/4.

## Exercise 16.4-4

**Exercise**

Suppose that instead of contracting a table by halving its size when its load factor drops below 1/4, you contract the table by multiplying its size by 2/3 when its load factor drops below 1/3. Using the potential function

$$\Phi(T) = |2(T.num - T.size/2)|$$

show that the amortized cost of a `TABLE-DELETE` that uses this strategy is bounded above by a constant.