Algorithmen und Datenstrukturen
SS 2017
Blatt 3

Antonio Rodriguez-Sanchez, Sebastian Stabinger, Simon Hangl

2017–04–06

Exercise 1 : Recursion

A function f on positive integers is defined recursively in the following way:

$$f(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ f(3n+1) & \text{if } n \text{ is odd} \\ n & \text{if } n = 4 \end{cases}$$

- Write a recursive implementation of this function in C, C++, or Java
- Determine the call tree (Aufrufbaum) for $f(14)$
- Rewrite the function in a non-recursive way

Optional Questions:

- Can you show that this function, given an integer, will always return 1 at some point?

Exercise 2 : Stack

- Implement a **stack for integers** in C, C++, or Java using **linked lists**. The following functions should be implemented.
  - `push` : Put an element on top of the stack
  - `pop` : Take the top most element from the stack and return it
  - `size` : Return the number of elements on the stack
- What is the runtime complexity of your implemented functions?
- Don't forget to test your Implementation on a few examples.

Exercise 3 : Queue

- Implement a **queue for integers** in C, C++, or Java using **linked lists**. The following functions should be implemented.
  - `enqueue` : Put an element at the end of the queue
  - `dequeue` : Take the first element from the queue and return it
  - `size` : Return the number of elements in the queue
- What is the runtime complexity of your implemented functions?
- Don't forget to test your Implementation on a few examples.

Exercise 4 : Queue

Suppose an initially empty queue Q has performed a total of 32 `enqueue` operations, 10 `first` operations, and 15 `dequeue` operations, 5 of which returned `null` to indicate an empty queue. What is the current size of Q?