Homework Assignment 1

Due Date: February 25, 2022, 23:59

EXERCISE 1. For $k \in \mathbb{N}, k \geq 1$ define an operation *delete_last*(ℓ, k), which deletes the last k elements in a list ℓ. Analyse the amortised complexity of this operation and show that it is in $\Theta(1)$, independent of k.

total points: 8

EXERCISE 2. Explore structural recursion on list objects, i.e. define an operation $src[e, f, g]$ in the following way:

- If ℓ is the empty list, then $src[e, f, g](\ell) = e$, where $e \in T'$ is some constant.
- If ℓ is a singleton list containing just one element x, then $src[e, f, g](\ell) = f(x)$, where f is a function that maps elements of a set T (the set of list elements) to elements of a set T'.
- If ℓ can be written as the concatenation of two lists, say $\ell = concat(\ell_1, \ell_2)$, then

$$src[e, f, g](\ell) = g(src[e, f, g](\ell_1), src[e, f, g](\ell_2)) \ ,$$

  i.e. apply structural recursion to both sublists separately, then apply the operation $g : T' \times T' \to T'$ to the resulting pair.

(i) Discuss the conditions, under which $src[e, f, g]$ is well-defined.

(ii) Show how to use structural recursion to define operations on lists such as

- determining the length,
- applying a function to all elements of a list, and
- creating a sublist of list elements satisfying a condition φ.

(iii) Analyse the time complexity of structural recursion.

total points: 14

EXERCISE 3. Explain how to implement a FIFO queue using two stacks so that each FIFO operation takes amortised constant time.

total points: 8

EXERCISE 4.

(i) Implement the function *delete_last*(ℓ, k) from Exercise 1 on the class ALIST.

(ii) Implement structural recursion from Exercise 2 on the class ALIST.

total points: 20