

## Foundations of Computer Science – Supervision 3

**Question 1:** (a) Code an analogue of `map` and `filter` for sequences.

- (b) Create a lazy list of:
  - (i) even numbers
  - (ii) square numbers
  - (iii) integers that are not divisible by three
- (c) What happens if you run `filter (x -> x mod 2 = 1)` on the sequence of even numbers? Why?
- (d) Define a type representing rational numbers. Then define a lazy list containing positive rational numbers. A number may appear in the sequence multiple times, but all rational numbers should be in the sequence.
- (e) (Optional) Define a lazy list containing all finite sets of integers. Is it possible to define a lazy list containing all sets of integers?

**Question 2:** (a) Compare depth-first search, breadth-first search and iterative deepening. What factors decide which technique should be used in a certain scenario?

- (b) A lazy binary tree is either empty or is a branch containing a label and two lazy binary trees, possibly to infinite depth. Present an OCaml datatype to represent lazy binary trees, along with a function that accepts a lazy binary tree and produces a lazy list that contains all of the tree's labels. The order of the elements in the lazy list does not matter, as long as it contains all potential tree nodes.
- (c) Define a datatype representing lazy (general) trees, where nodes may have arbitrarily many (possibly infinitely many) children.
- (d) How would you traverse a tree described in part (c)?
- (e) (Optional) Define a lazy tree with arbitrarily long paths (i.e. for any integer  $n$ , there exists a path of length  $n$ ), that has no infinite-length path.