Sequences (lazy lists) and trees are fundamental types in functional programming. Here are definitions of sequences and trees with integer elements:

\[
\begin{align*}
type \text{iseq} &= \text{Nil} \\
&\quad \mid \text{Cons of int * (unit -> iseq)} \\

type \text{itree} &= \text{Leaf of int} \\
&\quad \mid \text{Branch of itree * itree}
\end{align*}
\]

(a) In an ascending sequence such as 1, 3, 3, 7,\ldots each element is at least as large as the previous elements.

Given two ascending sequences, write a function \text{merge2} that produces a sequence of the elements of both in ascending order. For example, passing 1, 3, 3, 7,\ldots and 2, 4, 5, 9,\ldots to \text{merge2} should produce the sequence 1, 2, 3, 3, 4, 5, 7, 9,\ldots.

(b) Sequences are considered to be equal if corresponding elements are equal.

(i) Define a function \text{equal_seq} that compares two sequences for equality.

(ii) Define sequences \text{s1} and \text{s2} for which \text{equal_seq \ s1 \ s2} does not terminate.

(c) The fringe of a tree is the left-to-right sequence of the values at the leaves. For example, the fringe of \text{Branch (Leaf 3, Branch (Leaf 10, Leaf 4))} is the sequence 3, 10, 4.

(i) Define a function \text{fringe} that computes the fringe of a tree. Your function should have the following type:

\[
\text{val fringe : itree -> iseq}
\]

(ii) Using the functions you have defined above or otherwise, write a function \text{equal_fringes} that determines whether two trees have equal fringes.