Version 1.1 ***due:*** *Mon, Feb 4 @ 11:59pm*

1 Introduction

Task 1.1 (50%).

```
functor MkSkyline(structure S : SEQUENCE) : SKYLINE =
struct
  structure Seq = S
  open Seq

  fun unzip s =
      (map (fn (a,_) => a) s, map (fn (_,b) => b) s)

  fun skyline buildings =
      let
        fun addXs toAdd (xs, hs : int seq) =
            let
              fun copy ((_, SOME h), (x, NONE)) = (x, SOME h)
                | copy (_, r) = r
              val copyScan = scani copy (0, NONE)

              val xs' = map (fn x => (x, NONE)) toAdd
              val hs' = map2 (fn (x,h) => (x, SOME h)) xs hs
              fun cmpX ((x1,_), (x2,_)) = Int.compare (x1,x2)
            in unzip (copyScan (merge cmpX xs' hs'))
            end

        fun combine ((xs1, hs1), (xs2, hs2)) =
            let
              fun optMax ((NONE, x) | (x, NONE)) = valOf x
                | optMax (SOME x, SOME y) = Int.max (x, y)

              val (xs, hs1') = addXs xs2 (xs1, hs1)
              val (_,  hs2') = addXs xs1 (xs2, hs2)
            in (xs, map2 optMax hs1' hs2')
            end

        fun init (l,h,r) = (%[l,r], %[h, 0])
        val base = (empty (), empty ())
        val (xs, hs) = reduce combine base (map init buildings)
        fun isUniq (0, _) = true
          | isUniq (i, (x,h)) = nth hs (i-1) <> h

      in filterIdx isUniq (zip xs hs)
      end
end
```

Task 1.2 (10%).

We do not provide a reference solution for testing.

Task 1.3 (30%).

Let A be the following algorithm:

Given an empty sequence, return an empty sequence.

Given a sequence of length 1, return the point at the leftmost side and height of the building, and another point at the right side of the building at height 0.

Given a sequence of length greater that 1, split the sequence in half and perform A on each half, then comine the two results as follows:

Merge the two sequences into a new sequence sorted by x values with each point tagged with which sequence it came from. Next create two new sequences, one where each point from the right sequence has its height replaced by the height of the closest point from the left sequence to the left of it, and the other with the same thing, but the left height replaced by right heights instead. Then combine these two sequences into a single sequence by choosing the maximum height between the two sequences at each x.

Claim: If performing A on a sequence of buildings s results in s', then the x-coordinates expressed in s' are exactly the x-coordinates expressed in s, and for every $(x_i, y_i) \in s'$, y_i is the height of the tallest building $b = (l, h, r)$ for which $l \leq x_i < r$.

Proof by structural induction on `showt s`:

Case for `EMPTY`: The claim is true for the empty sequence because there are no x-coordinates in s or s'.

Case for `ELT b`: The claim is true when s contains a single building because the x-coordinates for the two points in s' are the same as the right and left side of `b`. Also, the height of the first point falls inside the range of `b` and has the same height as `b`, and the height of the second point does not fall inside the range of `b` and has height 0.

Case for `NODE (l, r)`:

I.H.: Assume that the claim holds for `l` and `r` with results l' and r'.

I.S.: Since A merges the two sequences to get all of the x-coordinates, and l' and r' have all of the x-coordinates in `l` and `r` respectively by the I.H., s' has all of the x-coordinates.

Since A chooses the highest height at each x-coordinate between l' and r', and by the I.H., l' and r' have the height of the highest building at that x-coordinate within `l` and `r`. eah y value in s' will meet the claim.

Since all of the cases hold, the claim is true for all sequences of buildings s.

Task 1.4 (10%).

Coming soon...