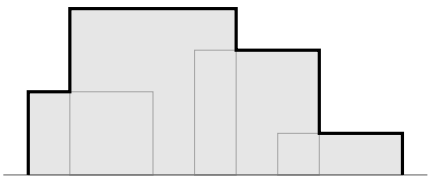
PSO 2 — Week 11: Problems

CS381, Fall 2026

Problem 1. Skylines

You are given n rectangular buildings standing on a horizontal line: building i covers the x-interval $[\ell_i, r_i]$ and has height $h_i > 0$. The *skyline* is the outline of all the buildings together, drawn thick in the picture. We represent it as a list of breakpoints $(x_1, y_1), \ldots, (x_k, y_k)$ with $x_1 < \cdots < x_k$, meaning that the skyline has height y_t from x_t until x_{t+1}; it has height 0 before x_1, and $y_k = 0$. In the picture the list is $(0, 2), (1, 4), (5, 3), (7, 1), (9, 0)$.



Give an $O(n \log n)$ divide-and-conquer algorithm that computes the skyline: split the buildings into two halves, find each half's skyline recursively, and merge the two skylines. Explain why the merge takes only linear time, and why the algorithm is correct.

Problem 2. Monotonic stacks

Let $A[1..n]$ be an array of distinct numbers.

(a) For each index i, let next(i) be the smallest index $j > i$ with $A[j] > A[i]$, or "none" if there is no such j. Compute next(i) for every i in $O(n)$ time using a stack. State the invariant your stack maintains, and explain why the total time is $O(n)$ even though a single step can pop many elements off the stack. (Hint: which indices are still waiting for their answer?)

(b) A histogram is a row of n bars of width 1 standing side by side, with heights $h_1, \ldots, h_n \geq 0$ (the heights need not be distinct). Find the largest area of a rectangle with horizontal and vertical sides that fits inside the histogram, in $O(n)$ time. Start by showing that some largest rectangle is exactly as tall as the shortest bar it covers. Once you know which bar that is, how far left and right can the rectangle reach?