In-class exercises: Divide-and-conquer

The skyline problem/the upper envelope problem:

In this problem we design a divide-and-conquer algorithm for computing the skyline of a set of n buildings.

A *building* B_i is represented as a triplet $(\mathbf{L_i}, H_i, \mathbf{R_i})$ where $\mathbf{L_i}$ and $\mathbf{R_i}$ denote the left and right x coordinates of the building, and H_i denotes the height of the building (note that the x coordinates are drawn boldfaced.)

A *skyline* of a set of n buildings is a list of x coordinates and the heights connecting them arranged in order from left to right (note that the list is of length at most $4n$).

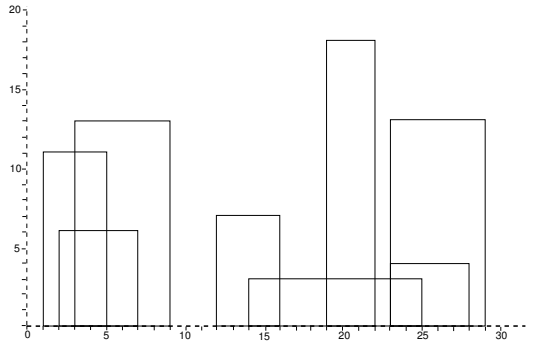
Example: The skyline of the buildings

$$\{(\mathbf{3}, 13, \mathbf{9}), (\mathbf{1}, 11, \mathbf{5}), (\mathbf{12}, 7, \mathbf{16}), (\mathbf{14}, 3, \mathbf{25}), (\mathbf{19}, 18, \mathbf{22}), (\mathbf{2}, 6, \mathbf{7}), (\mathbf{23}, 13, \mathbf{29}), (\mathbf{23}, 4, \mathbf{28})\}$$

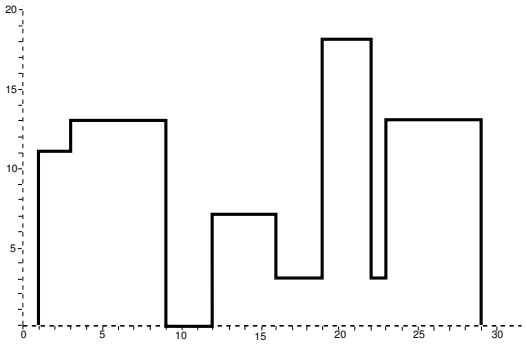
is

$$\{\mathbf{1}, 11, \mathbf{3}, 13, \mathbf{9}, 0, \mathbf{12}, 7, \mathbf{16}, 3, \mathbf{19}, 18, \mathbf{22}, 3, \mathbf{23}, 13, \mathbf{29}, 0\}$$

(note that the x coordinates in a skyline are sorted).



buildings



skyline

1. Let the size of a skyline be the total number of elements (coordinates and heights) in its list.

   Describe an algorithm for combining a skyline A of size n_1 and a skyline B of size n_2 into one skyline S of size $O(n_1 + n_2)$. Your algorithm should run in time $O(n_1 + n_2)$.

2. Describe an $O(n \log n)$ algorithm for finding the skyline of n buildings.