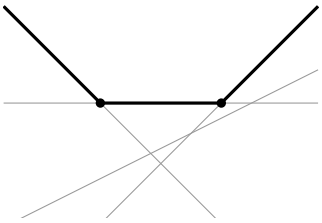
PSO 2 — Week 7: Problems

CS381, Fall 2026

Problem 1. Upper envelopes

You are given n lines, the ith being the graph of $y = a_i x + b_i$. Assume the slopes are distinct and no three lines pass through a common point. The *upper envelope* is the function $U(x) = \max_i (a_i x + b_i)$: what you see looking down on the lines from above. It is made of pieces, each lying on one line, and we represent it by listing the lines of the pieces from left to right, together with the x-coordinates where one piece ends and the next begins. In the picture, the envelope of four lines is drawn thick, and one of the four never reaches the top.



(a) Sort the lines by slope and add them one at a time in that order, keeping the lines of the current envelope on a stack, ordered left to right from the bottom of the stack to the top. When a new line arrives, say exactly when the line on top of the stack should be popped. Prove the method correct, and show that everything after the sort takes $O(n)$ time. (Hint: the new line is the steepest so far, so it ends up on top for all large enough x.)

(b) Now give a divide-and-conquer algorithm instead: split the lines into two halves in any way, find each half's envelope recursively, and combine the two envelopes in linear time. What is the total running time? You may use the fact, which follows from (a), that each line appears on an envelope at most once, so an envelope of m lines has at most m pieces.