We first label the lines in order of increasing slope, and then use a divide-and-conquer approach. If $n \leq 3$ — the base case of the divide-and-conquer approach — we can easily find the visible lines in constant time. (The first and third lines will always be visible; the second will be visible if and only if it meets the first line to the left of where the third line meets the first line.)

Let $m = \lceil n/2 \rceil$. We first recursively compute the sequence of visible lines among $L_1, \ldots, L_m$ — say they are $\mathcal{L} = \{L_{i_1}, \ldots, L_{i_p}\}$ in order of increasing slope. We also compute, in this recursive call, the sequence of points $a_1, \ldots, a_{p-1}$[1] where a_k is the intersection of line L_{i_k} with line $L_{i_{k+1}}$. Notice that $a_1, \ldots, a_{p-1}$ will have increasing x-coordinates; for if two lines are both visible, the region in which the line of smaller slope is uppermost lies to the left of the region in which the line of larger slope is uppermost. Similarly, we recursively compute the sequence $\mathcal{L}' = \{L_{j_1}, \ldots, L_{j_q}\}$ of visible lines among $L_{m+1}, \ldots, L_n$, together with the sequence of intersection points $b_k = L_{j_k} \cap L_{j_{k+1}}$ for $k = 1, \ldots, q-1$.

To complete the algorithm, we must show how to determine the visible lines in $\mathcal{L} \cup \mathcal{L}'$, together with the corresponding intersection points, in $O(n)$ time. (Note that $p + q \leq n$, so it is enough to run in time $O(p+q)$.) We know that L_{i_1} will be visible, because it has the minimum slope among all the lines in this list; similarly, we know that L_{j_q} will be visible, because it has the maximum slope.

We merge the sorted lists $a_1, \ldots, a_{p-1}$ and $b_1, \ldots, b_{q-1}$ into a single list of points $c_1, c_2, \ldots, c_{p+q-2}$ ordered by increasing x-coordinate. This takes $O(n)$ time. Now, for each k, we consider the line that is uppermost in $\mathcal{L}$ at x-coordinate c_k, and the line that is uppermost in $\mathcal{L}'$ at x-coordinate c_k. Let ℓ be the smallest index for which the uppermost line in $\mathcal{L}'$ lies above the uppermost line in $\mathcal{L}$ at x-coordinate c_ℓ. Let the two lines at this point be $L_{i_s} \in \mathcal{L}$ and $L_{j_t} \in \mathcal{L}'$. Let (x^*, y^*) denote the point in the plane at which L_{i_s} and L_{j_t} intersect. We have thus established that x^* lies between the x-coordinates of $c_{\ell-1}$ and c_ℓ. This means that L_{i_s} is uppermost in $\mathcal{L} \cup \mathcal{L}'$ immediately to the left of x^*, and L_{j_t} is uppermost in $\mathcal{L} \cup \mathcal{L}'$ immediately to the right of x^*. Consequently, the sequence of visible lines among $\mathcal{L} \cup \mathcal{L}'$ is $L_{i_1}, \ldots, L_{i_s}, L_{j_t}, \ldots, L_{j_q}$; and the sequence of intersection points is $a_{i_1}, \ldots, a_{i_{s-1}}, (x^*, y^*), b_{j_t}, \ldots, b_{j_{q-1}}$. Since this is what we need to return to the next level of the recursion, the algorithm is complete.

[1] ex428.913.582