CS325: Revisit Visibility Project

Prof. Borradaile

Due: Tuesday, October 28 at 10AM

Your report must be typeset, printed and stapled. *Each team member's name must be listed as well as any resources used to finish the project. For questions regarding this project, please contact your TA, Spencer Hubbard at* `hubbarsp@eecs.oregonstate.edu`.

For this project, you will revisit the visibility problem and design, analyze and implement a (hopefully) more efficient divide-and-conquer algorithm. Recall the visibility problem:

> Given lines $y_1, \ldots, y_n$ where $y_i(x) = m_i x + b_i$ and $m_1 < \ldots < m_n$, find the subset of visible lines. A line y_i is visible if there exists x such that for all indices j, $y_i(x) \geq y_j(x)$.

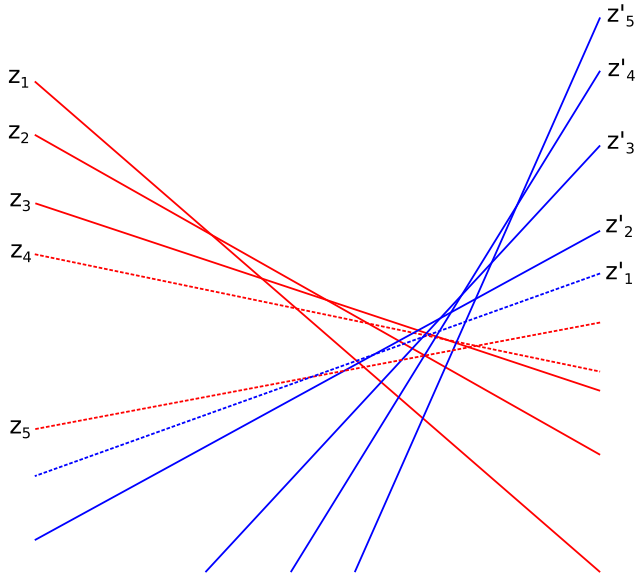
To get there, we will start with a warm-up problem, the merging visible lines problem.

Merging visible lines

Consider the following problem:

> Given two sets of lines $\{z_1, z_2, \ldots, z_t\}$ and $\{z'_1, z'_2, \ldots, z'_s\}$, each ordered by (strictly) increasing slope, and such that the slope of z_t is strictly less than the slope of z'_1, find the visible subset of $\{z_1, z_2, \ldots, z_t\} \cup \{z'_1, z'_2, \ldots, z'_s\}$.

Consider the following example. Among the red lines, z_1, z_2, z_3, z_4, z_5 are all visible and among the blue lines $z'_1, z'_2, z'_3, z'_4, z'_5$ are all visible. Among $z_1, z_2, z_3, z_4, z_5, z'_1, z'_2, z'_3, z'_4, z'_5$ the solid lines are visible: $z_1, z_2, z_3, z'_2, z'_3, z'_4, z'_5$.



Can you see a pattern? In fact, there is a pattern; the following is true (as you will prove):

Claim 3: If $\{z_1, z_2, \ldots, z_t\}$ and $\{z'_1, z'_2, \ldots, z'_s\}$ Are two visible set of lines (each ordered by increasing slope), then the visible subset of $\{z_1, z_2, \ldots, z_t\} \cup \{z'_1, z'_2, \ldots, z'_s\}$ is $\{z_1, \ldots, z_i\} \cup \{z'_j, \ldots, z'_s\}$ for some $i \geq 1$ and $j \leq s$.

Given this claim, how do you *quickly* find i and j? Suppose you have discovered that $\{z_1, \ldots, z_k\} \cup \{z'_\ell, \ldots, z'_s\}$ is visible. Can you easily determine if $\{z_1, \ldots, z_{k+1}\} \cup \{z'_\ell, \ldots, z'_s\}$ or $\{z_1, \ldots, z_k\} \cup \{z'_{\ell-1}, \ldots, z'_s\}$ are visible? We will call this algorithm **MergeVisible**.

Be sure to test (experimentally) that your implementation of **MergeVisible** is correct! You can use your implementations from Project 1 as a point of comparison.

Divide and Conquer

You will use **MergeVisible** to help design a divide-and-conquer algorithm **Algorithm 4** for the original visible lines problem. If you want to determine the visible subset of $\{y_1, \ldots, y_n\}$, you can recursively compute the visible subsets of $y_1, y_2, \ldots, y_\ell$ and $y_{\ell+1}, y_{\ell+2}, \ldots, y_n$ for $\ell = \lfloor n/2 \rfloor$. These recursively computed subsets can then be merged using **MergeVisible**. Note that you should design **MergeVisible** so that the running time of **Algorithm 4** is better than **Algorithm 3** (from Project 1).

Project report

Your report **must** include:

Proof of Claim 3 You can use any claims from previous handouts, if they are helpful.

Run-time analysis Give pseudocode for MergeVisible and Algorithm 4 and an analysis of the asymptotic running-time of each algorithm.

Proofs of Correctness Give a proof by induction that Algorithm 4 returns the correct solution. You may use the fact that Claim 3 is true.

Experimental correctness To illustrate that your code is correct, determine the solution for the single instance in the file:`http://web.engr.oregonstate.edu/~glencora/cs325/visibility/solve_these_2.txt` The format of this file is the same as in Project 1. You should submit a solution file to TEACH as in Project 1 with the filename `proj2_grp<i>.txt` where $< i >$ is your group number..

Experimental analysis Perform an experimental analysis of Algorithm 4 as described in the first project. For your plots, include the data collected for Algorithms 1, 2 and 3 that you performed in the first project. An easy way to generate an instance of size 900 is to use the slopes $-449, -448, \ldots, 448, 450$ and then use randomly selected (integer) intercepts (over a large enough domain).

Extrapolation and interpretation Use the data from the experimental analysis to answer the following questions:

1. For Algorithm 4, what is the size of the biggest instance that you could solve with your algorithm in one hour? How does this compare with your answers from the first project?
2. Determine the slope of the line for Algorithm 4 in your log-log plot and from these slopes infer the experimental running time for these algorithms. Discuss any discrepancies between the experimental and theoretical running times.

Code Upload your code to T.E.A.C.H. Only one student from each group should do this.

Note!
1. You must use the same language you chose for the first project.
2. Be sure that you never use division as in the first project!