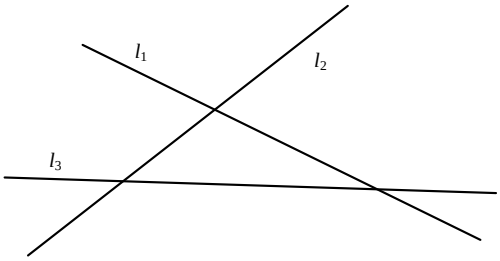
IT 427, Design and Analysis of Algorithms

Programming Assignment 5: Closest Pair

Due date: Oct. 23, 2024, Wednesday, 11:59 PM 50 points (35 on programs, 15 on report)

A line is represented by two parameters m and k as $y = mx + k$. Let $(m_1, k_1), (m_2, k_2), \ldots, (m_n, k_n)$ represent n lines where no three lines meet at a same point. A line $y = m_i x + k_i$ is said to be visible from above if there is a point (a,b) on the line, i.e, $b = m_i a + k_i$ such that, $b > m_j a + k_j$ for all j with $1 \leq j \leq n$ and $i \neq j$. For example, in the following figure, ℓ_1 and ℓ_2 are visible, while ℓ_3 is invisible.



For a variety of applications in computer graphics, we need to find visible lines quickly. This assignment asks you to use divided-and-conquer technique to find all visible lines from a given set of lines in time complexity $O(n \log n)$.

Prepare your programs on Linux Server: This is similar to the previous assignments.

- Make a directory `asg5` under your `IT427`, i.e., ∼`/IT427/asg5/`. All of your programs and needed files should be saved under your ∼`/IT427/asg5` before run `submit427.sh`.
- Check the contents of my `/home/ad.ilstu.edu/cli2/Public/IT427/asg5`. There are 6 text files, where each is a set of straight lines. You can follow the same format and create your own test points. I will test your program on a different sets of points.

Input: The input text file contains a set of straight lines in the form $y = mx + k$, as shown below. Except for the first line, each subsequent line represents one straight line in the format $y = mx + k$, where the first number is m and the second number is k.

```
--------------
     -3.9595310627     -7.9972257780
      0.2263071750      2.7253545171
      0.4716764577     -5.1356486074
     ...               ...
     ...               ...
```

Output: I will run your program and expect the out as follows:

```
python3 visibleLines.py lineSet1.txt

0:  m:  -3.95953 k:   -7.99723
1:  m:  -0.54274 k:   45.71587
2:  m:  -0.14801 k:   40.35810
3:  m:   2.69344 k:  -43.20543


by Chung-Chih Li
```

Important Notes: The score of your program not only depends on the correctness of your program, but also on the efficiency of your program. If your program runs at $O(n^2)$, you will not get more than 50% of the score (on both program and report parts). Also, I will expect to see time complexity analysis with necessary mathematical proof and arguments to explain how to get $O(n \log n)$. That includes well-defined recurrence relation based on your divide-and-conquer algorithm with sufficient explanation. Without adequate arguments in your report, you will get at most 10 points on the report.

Submission: Programs (35 points) and Reports (15 points)

Submission details are same as the previous assignment. Run the submission script with the submission number changed to 6, but you can use the same secret name as follow:

```
bash /home/ad.ilstu.edu/cli2/Public/IT427/submit427.sh peekapoo 5
```

Note that, since I will keep updating `submit427.sh` for different assignment, you have to run the script from my `/home/ad.ilstu.edu/cli2/Public/IT427/` directly for the most recent updated version, i.e., don't copy it to your own directory.

1. Programs: 35 points. Submission on Linux server.
   The score is based on the correctness and the programming style, which includes efficiency, appropriateness of data structures, and documentation of your programs. At the beginning of every program file, put a section of comments including (1) your full name, (2) student ID, (3) a pledge of honesty that you do not copy/modify from other's codes and (4) a declaration of your copyright that no one else should copy/modify the codes. You will receive:
   (a) 95 $\sim$ %: No error with a good programming style.
   (b) 80 $\sim$ %: Minor error and fair programming style.
   (c) 60 $\sim$ %: Some error and not so good but acceptable programming style.
   (d) 40 $\sim$ %: Too many error and bad programming style, but meaningful.
   (e) 20 $\sim$ %: Compilable but not working and the program must show reasonable trying.
   (f) 0 $\sim$ %: : Fail to meet any of aforementioned qualities or plagiarism involved.
2. Report: 15 points. Submission through Canvas.
   You have to write up a report and prepare it in pdf format. You don't have to put program output on the report as I will run and exam your program directly on some different input files.
   The report should includes brief descriptions of your program, summary of the methods, data structures, and efficiency analysis on time and space in details in terms of big-O notations. If there is any difficulties encountered in this assignment, you can report it. If your analysis is not clearly related to your program with sufficient justification, your report score will not be higher than 50%.