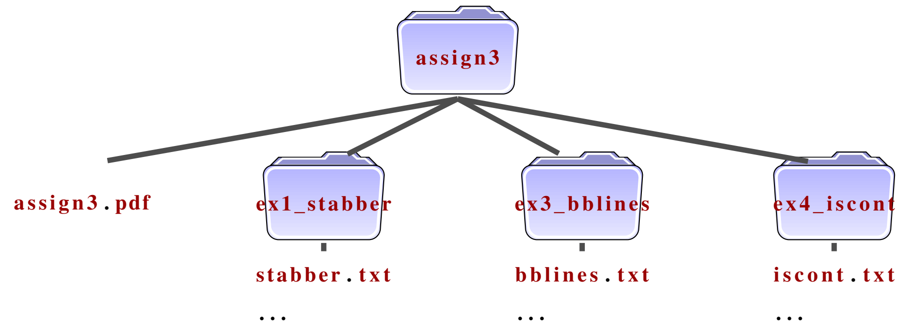
Assignment no. 3

due: Tuesday 24th January, 2023, midnight

Before starting to answer the questions please read them very carefully. Rigorously follow the submission guidelines. If appropriate, your programs should detect invalid input data and print out relevant error messages. Do not add "friendly" messages to your programs. You may work on and submit this assignment **in pairs**.

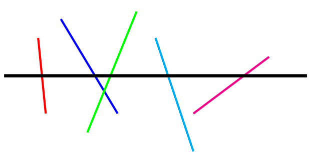
Please provide an archive file, the basename of which is `assign1`, that contains all source and documentation files *and only these files* in a 2-level directory structure:



The archive should contain a pdf file called **assign3.pdf** at the highest level. This file should contain the answers to **all** exercises.

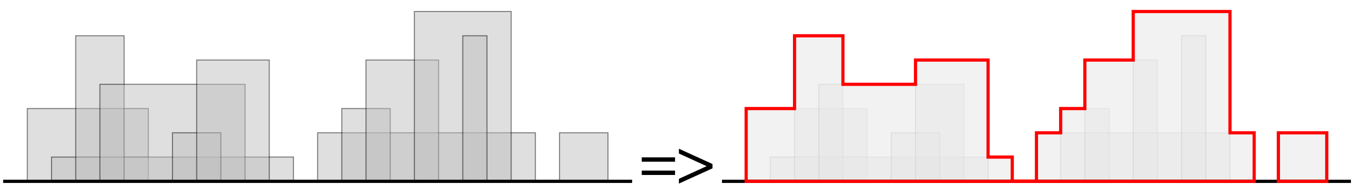
If you develop in C++, for each programming exercise your solution should consist of one source file (the base-name of which matches the name of the executable specified in the exercise), **CMakeLists.txt** (input to **cmake**), and a pdf or text file with extension **.txt** (the base-name of which matches the name of the executable specified in the exercise) that **only** provides implementation details and instructions for running the program. Running **cmake** on the provided **CMakeLists.txt** followed by compilation and linkage should produce the desired executable. If you develop in Python, for each exercise provide a script (the base-name of which matches the name of the executable specified in the exercise) as well as a pdf or text file as described above.

3.1. (CGAA Ex. 8.16) Let S be a set of n closed segments in the plane. A line ℓ that intersects all segments of S is called a transversal or stabber for S.



Give an $O(n^2)$ algorithm that decides whether a stabber exists for S and find a representative. Implement your algorithm as a program called **stabber**.

3.2. Given n axis-parallel rectangles in the plane with their bottom sides lying on the x-axis. Devise an algorithm that constructs the union of the rectangles in $O(n \log n)$ time.



See next page

3.3. Let L be a set of n lines in the plane. Give an $O(n \log n)$ time algorithm to compute an axis-parallel rectangle that contains all the vertices of the arrangement $\mathcal{A}(L)$ in its interior. Implement your algorithm as a program called **bblines**.

3.4. Hopcroft's problem is to decide, given n lines and n points in the plane, whether any point is contained in any line. Give an $O(n^{3/2} \log n)$ time algorithm to solve Hopcroft's problem. Hint: Give an $O(n \log n)$ time algorithm to decide, given n lines and $\sqrt{n}$ points in the plane, whether any point is contained in any line.

Implement your algorithm as a program called **iscont**. Fine tune your implementation. Your implementation will be benchmarked for time efficiency.

3.5. (CGAA Ex. 3.17) In class we gave a $2n$ upper bound on the complexity of the union of a set of polygonal pseudodisks with n vertices in total.

- Prove an upper bound of $2n-3$ on the complexity of the union.
  Hint: show that the complexity of the union is at most $2n-m$, where m is the number of original vertices in the union.
- Prove a lower bound of $2n-6$ by constructing an example that has this complexity.

3.6. Prove the following theorem:

Theorem. *Let H be a hole in polygon P. Then, $P \oplus Q \neq (P \cup H) \oplus Q$ iff $\exists t \in \mathbb{R}^2$, such that $Q \oplus \{t\} \subseteq -H$.*

Hint: First prove that $P \cap (Q \oplus \{t\}) \neq \emptyset$ iff $t \in P \oplus -Q$, where $\hat{P} = P \cup H$ denote the polygon P with the hole H filled up.