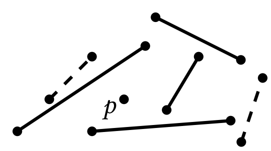
Computational Geometry
Summer Term 2025
scale.iti.kit.edu

Exercise Sheet 2

Submission due by 2025-05-29

Problem 1: Dangerous Walls — 5 points

A new streetlamp has been installed at point p on campus. Since then, staff have increasingly complained that their offices are too bright. As a result, n infinitely high (and infinitely thin) straight walls have been erected to shield certain offices from the light. However, unlit walls pose a safety hazard for pedestrians and aircraft, so walls that are completely in the shadow of others must be dismantled. Describe an efficient algorithm that determines which walls are dangerous.



Problem 2: Triangulation — $2 + 2 + 2 = 6$ points

Part (a) Let P be a simple polygon with n vertices. There may be multiple ways to triangulate P. Into how many triangles is P partitioned by any of these triangulations? *(Prove your answer!)*

Part (b) Now consider a polygon P that may contain holes. Prove that P can always be triangulated.

Part (c) Let P be a polygon that may contain holes. The boundary of the polygon is defined by n vertices. There may be multiple ways to triangulate P. Into how many triangles is P partitioned by any of these triangulations? *(Prove your answer!)*

Hint: In a planar graph with n vertices and m edges, it holds that $m \leq 3n - 6$. Equality holds if and only if the boundary of every face (including the outer one) is a triangle.

Problem 3: y-Monotone Triangulation — 5 points

Design an algorithm that, given a y-monotone polygon P with n vertices as input, outputs the diagonals that triangulate P in $\mathcal{O}(n)$ time. Show that your algorithm is correct and respects the claimed time complexity.

Problem 4: Completing 2D Linear Programming **4 points**

Give an algorithm that, given a 2-dimensional linear program (LP) with n constraints, determines whether it is unbounded in $\mathcal{O}(n)$ time. How can your algorithm be used to complete the 2D LP algorithm presented in the lecture?