

DM819 – Computational Geometry

(Geometriske Algoritmer)

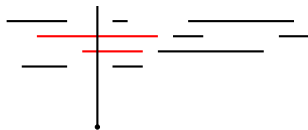
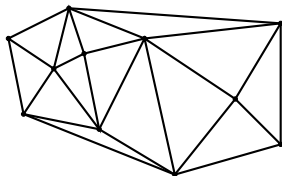
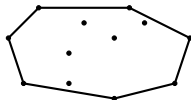
“Elective Courses Presentation”

Kim Skak Larsen

IMADA
University of Southern Denmark

`kslarsen@imada.sdu.dk`

`imada.sdu.dk/u/kslarsen/`



- ▶ Given n points, compute the *convex hull*. Be able to maintain it dynamically under insertion of points.
- ▶ Given n points, compute their *triangulation*.
- ▶ Preprocess n line segments in $O(n \log n)$, building a data structure in preparation for *ray shooting* so that queries such as this can be answered in time $O(\log n + k)$, where k is the number of line segments to report.

Algorithms and **Data Structures** for **Geometric Objects**

- ▶ Triangulations and Voronoi Diagrams
- ▶ Interval and Point Searches
- ▶ Convex Hulls
- ▶ Ray Shooting and Range Searching
- ▶ Motion Planning and more...

Introduction to (continuation of) important general techniques:

- ▶ Plane Sweeping and Fractional Cascading
- ▶ Randomization and Amortization

Geometric algorithms have applications (mostly not covered) in

- ▶ Computer Graphics, Geographic Information Systems, Design
- ▶ Robot Motion Planning, Image Analysis, Computer Games

Analysis of Algorithms and Data Structures

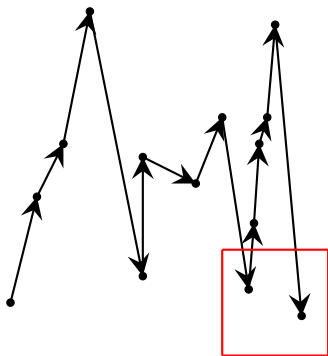
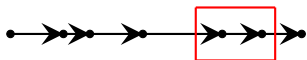
- ▶ primarily *algorithms and data structures*
- ▶ and a little *discrete math* and *probability theory*
- ▶ for maturity, having followed *advanced algorithms* and *computability and complexity* wouldn't hurt

More concretely,

- ▶ Search Trees (red-black trees) and Priority Queues
- ▶ Divide and Conquer
- ▶ Asymptotic Notation
- ▶ Correctness Analysis
- ▶ Time and Space Analysis,
including simple probability theory and adversarial arguments

If you want the course, but don't have the prerequisites, then talk to me!

Why Is This a Separate Course?



Q: It appears that it's basically algorithms! Why is it then a separate course and not just a part of other courses?

A: Beyond one dimension, there's no total ordering that preserves geometric proximity - this leads to all sorts of (fun) challenges that are quite different from the usual set-ups!

- ▶ 10 ECTS over one semester
- ▶ Exam: oral (highest weight) and programming project (in parts)
- ▶ 2h lectures, 2h exercises per week
- ▶ Programming project continuously throughout the semester
- ▶ Good book: Computational Geometry
Algorithms and Applications, 3. eds.
de Berg, Cheong, van Kreveld, Overmars
Springer, 2008.

- ▶ Chalk & blackboard lectures
- ▶ Core algorithmic problems (not graphics and games)
- ▶ The focus is on *efficient* algorithms – mostly $O(n \log n)$
- ▶ It says “geometry”, but this is *not* math! (\leq high school geometry)
- ▶ But there will be proofs/arguments in most lectures (of correctness and/or complexity)
- ▶ Course language is English, if necessary
- ▶ For the project, I expect that you can debug your own programs

See My Home Page for More Information

