Karlsruhe Institute of Technology

Geoinformatics A - Exercise

Part II
Introduction into Algorithmic Problems

Schedule

- ~~Week 1 Introduction into SQL~~
- ~~Week 2 Exercise I – Free Working Time~~
- Week 3 Introduction into Algorithmic Problems
- Week 4 Exercise II – Free Working Time
- Week 5 Exercise II – Free Working Time

Individual appointments can be arranged
if there are any problems!
Mail: ruiqi.liu9@kit.edu

Algorithm problems in GeoInfo

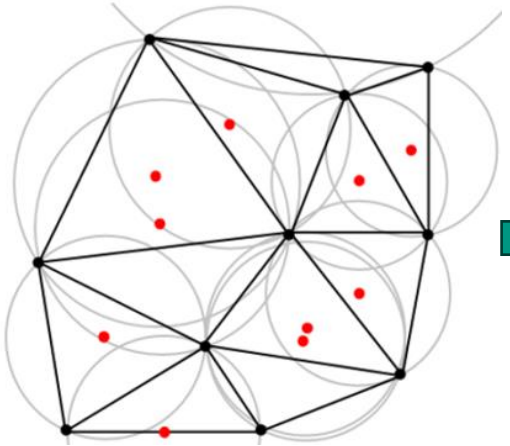
In Geoinformatics, algorithms are used to process, analyze, and manage spatial data, in order to present and understand spatial relationships from a geographic perspective.

- Triangulation
- Convex Hull
- Point in Polygon
- Nearest Neighbor

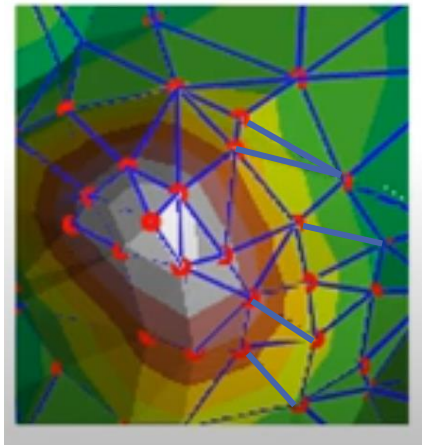
Algorithmic Problem I – Triangulation

A triangulation is a partition of a geometric domain, such as a point set, polygon, or polyhedron, into simplices that meet only at shared faces.

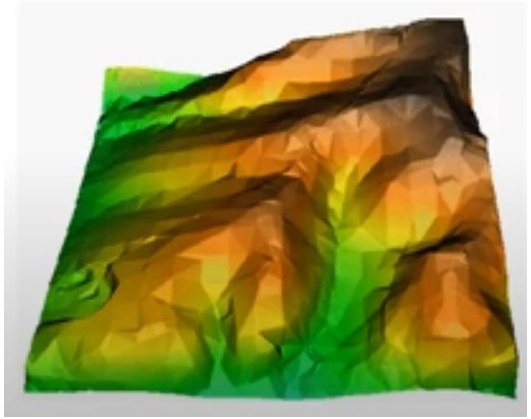
Irregularly elevation datasets



Triangulation



Terrain visualization

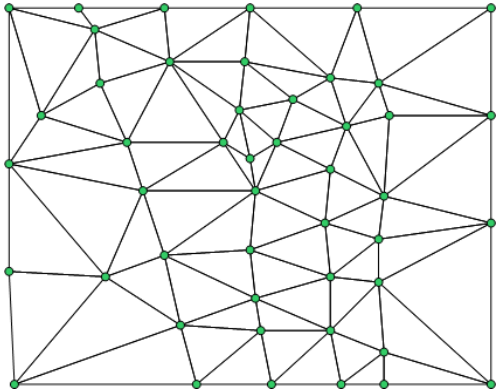


source: Mitas, L., Mitasova, H. (1999)

Algorithmic Problem I – Triangulation

Input:

- A set of 200 2D-Points with random X and Y coordinates within the interval [0, 10000]



Algorithm:

- Implementation of a triangulation on the generated point set

Output:

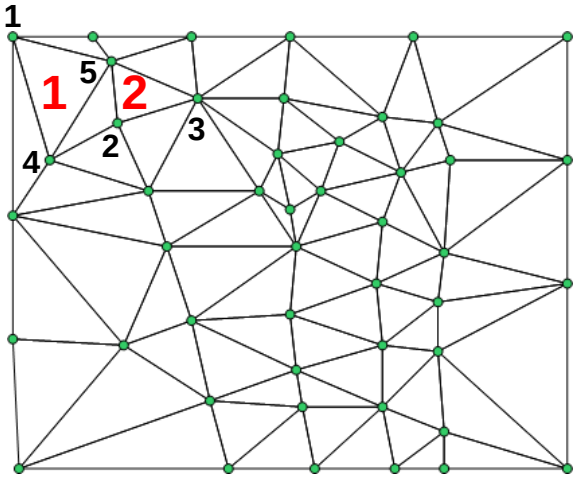
- Point List (Point Number, X-Coordinate, Y-Coordinate)
- Triangulation (Triangle Number, Point Number 1, Point Number 2, Point Number 3)

Algorithmic Problem I – Example

Output:

- Point List (Point Number, X-Coordinate, Y-Coordinate)

Point Number	X-Coordinate	Y-Coordinate
1	8401	5174
2	8967	8789
3	3772	985
4	4834	3915
5	6811	5730

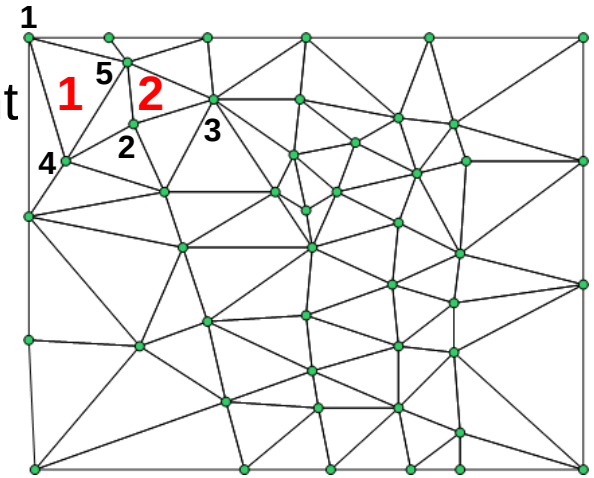


Algorithmic Problem I – Example

Output:

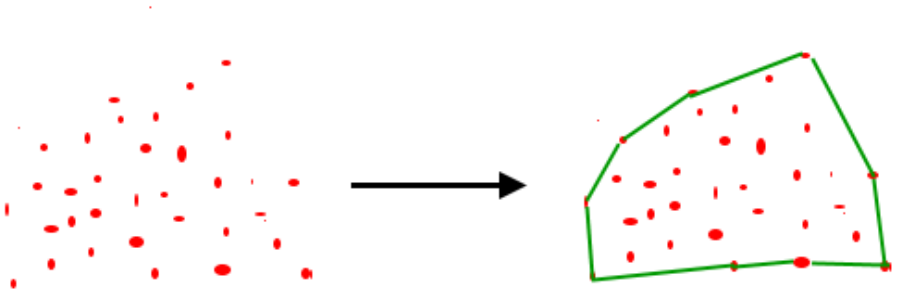
- Triangulation (Triangle Number, Point Number 1, Point Number 2, Point Number 3)

Triangle Number	Point Number 1	Point Number 2	Point Number 3
1	1	4	5
2	5	2	3



Algorithmic Problem II – Convex Hull

- Given a set of points “S” with (x, y) coordinates, the convex hull is the smallest convex set that contains all points in S.



source:https://en.wikipedia.org/wiki/Convex_hull_algorithms

Algorithmic Problem II – Convex Hull

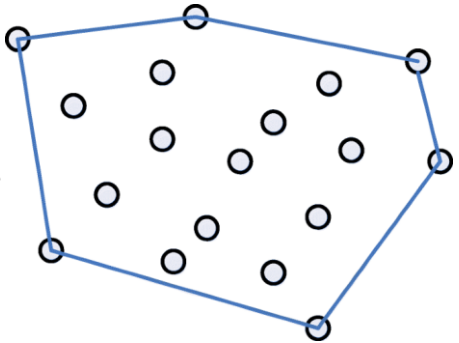
A convex hull around a polygon feature can be used to simplify the geometry of irregular shapes that may possess a significant number of vertices for processing. This is useful for generalizing shapes and improving storage performance.

source:https://telecommunications4dummies.com/2024/01/07/creating-areas-and-separating-points-based-on-its-location/

Algorithmic Problem II – Convex Hull

Input:

- A set of 200 2D-Points with random X and Y coordinates within the interval [0, 10000]



Algorithm:

- Determination of the Convex Hull of the set of 2D-Points

Output:

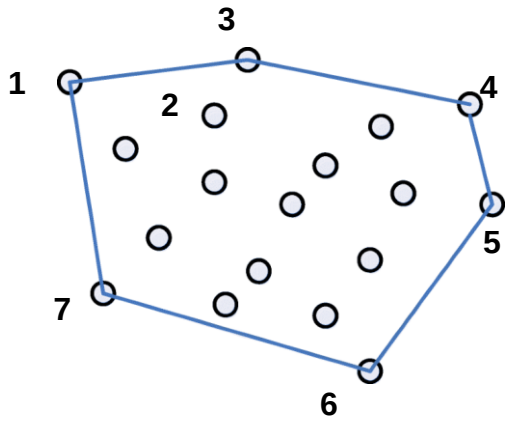
- Point List (Point Number, X-Coordinate, Y-Coordinate)
- Convex Hull (Point Numbers)

Algorithmic Problem II – Convex Hull

Output:

- Point List (Point Number, X-Coordinate, Y-Coordinate)

Point Number	X-Coordinate	Y-Coordinate
1	…	…
2	…	…
3	…	…
4	…	…
5	…	…
6	…	…
7	…	…

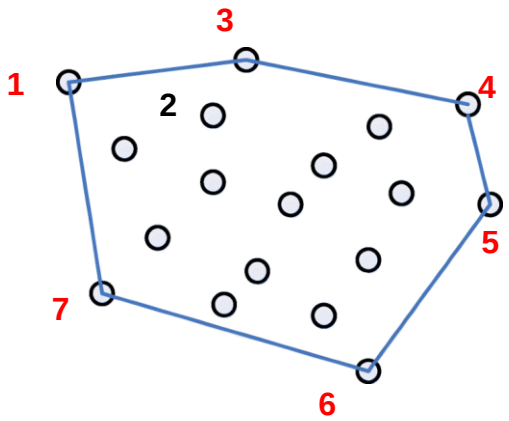


Algorithmic Problem II – Convex Hull

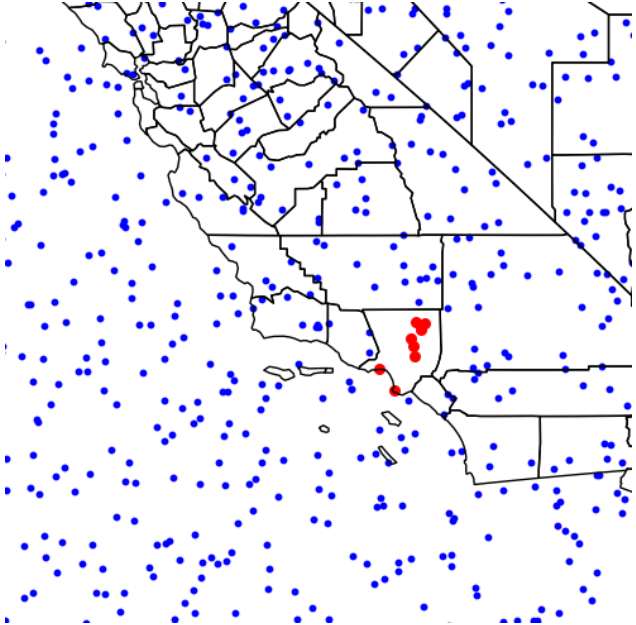
Output:

- Convex Hull (Point Numbers)

Convex Hull	Point Number
1	1, 3, 4, 5, 6, 7



Algorithmic Problem III – Point in Polygon



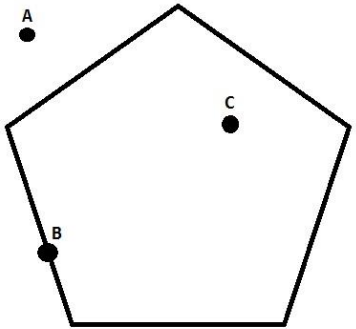
source:https://www.matecdev.com/posts/point-in-polygon.html#google_vignette

When you have a polygon layer and a point layer - and want to know how many or which of the points fall within the bounds of each polygon, you can use this method of analysis.

Algorithmic Problem III – Point in Polygon

Input:

- A predefined Polygon and a list of predefined 2D-Points
  - see Ilias



Algorithm:

- Read the Polygon and the list of Points
- Check if the points are within the Polygon or outside

Output:

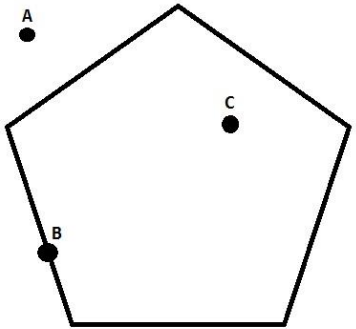
- Point List (Point Number, X-Coordinate, Y-Coordinate, True/False)

Algorithmic Problem III – Point in Polygon

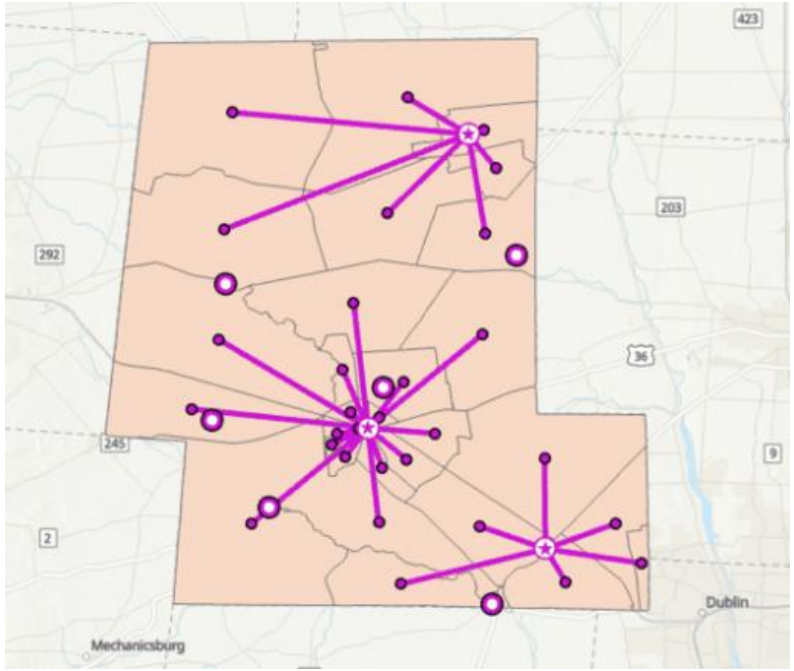
Output:

- Point List (Point Number, X-Coordinate, Y-Coordinate, True/False)

Point Number	X-Coordinate	Y-Coordinate	Point in Polygon
A	…	…	False
B	…	…	True
C	…	…	True



Algorithmic Problem IV – Nearest Neighbor



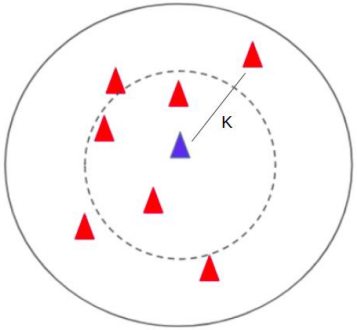
source:https://www.geographyrealm.com/basic-uses-of-gis/

Given a set of **n** points and query points, **q**, the nearest-neighbor (NN) problem is concerned with finding the point closest to the query points.

Algorithmic Problem IV – Nearest Neighbor

Input:

- Set of 200 2D-Points with random X and Y coordinates within the interval [0, 10000] (**A**)
- Set of 20 2D-Points with random X and Y coordinates within the interval [0, 10000] (**B**)



Algorithm:

- Determination of the nearest point in B for each point in A.

Output:

- Point List of A + B (Point Number, X-Coordinate, Y-Coordinate)
- Point Pairs (Point Number A with Coords., Point Number B with Coords.)

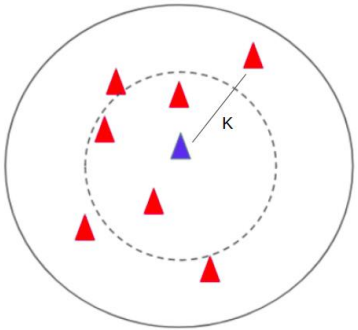
Algorithmic Problem IV – Nearest Neighbor

Output:

- Point List of A + B (Point Number, X-Coordinate, Y-Coordinate)

Point List of A

Point Number	X-Coordinate	Y-Coordinate
1	4036	469
2	19	3133
3	6806	885
…	…	…
200	…	…



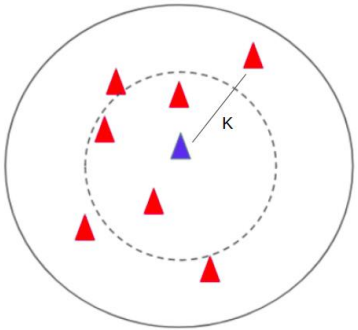
Algorithmic Problem IV – Nearest Neighbor

Output:

- Point List of A + B (Point Number, X-Coordinate, Y-Coordinate)

Point List of B

Point Number	X-Coordinate	Y-Coordinate
1	5129	6523
2	4806	5144
3	9507	3573
…	…	…
20	…	…



Algorithmic Problem IV – Nearest Neighbor

Output:

- Point Pairs (Point Number A with Coords., Point Number B with Coords.)

Point Pairs

Point in A	X-Coordinate	Y-Coordinate	Point in B	X-Coordinate	Y-Coordinate
1	4036	469	8	…	…
2	19	3133	10	…	…
3	6806	885	15	…	…
…	…	…	…	…	…
200	…	…	20	…	…

Project Setup

The following four tasks should be considered:

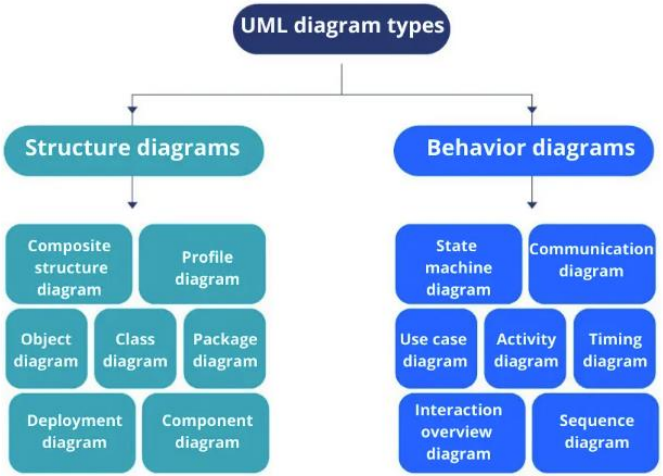
- Explanation of the problems (drawings, graphics, special cases, …)
- Discussion of the solution strategy (→ explaining the algorithm in words)
- Program conception (UML, pseudo code, …)
  - Representation of classes and interfaces as class diagrams (UML)
  - Representation of methods as activity diagram (UML)
- Implementation in any programming language (C++, Java, Python, …)

Project Setup

The following four tasks should be considered:

- UML (Unified Modeling Language)

Unified Modeling Language (UML) is a general-purpose modeling language. The main aim of UML is to define a standard way to visualize the way a system has been designed.

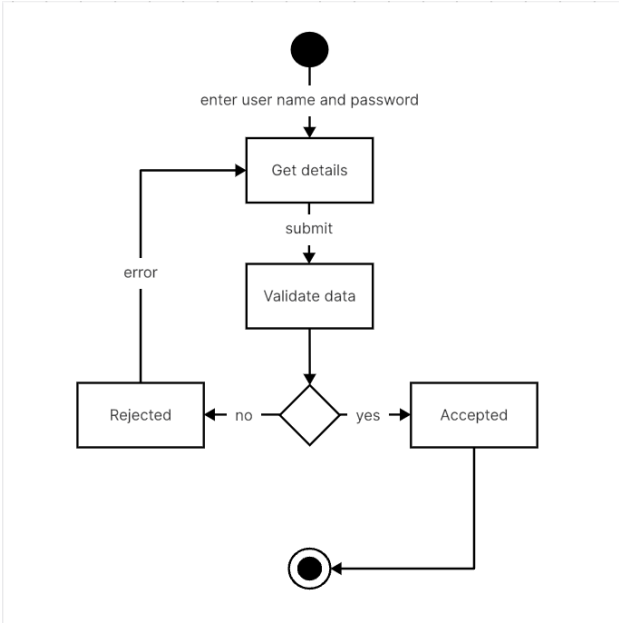


source: https://www.gleek.io/blog/uml-diagram-types

Project Setup

The following four tasks should be considered:

- UML (Unified Modeling Language)
  - Activity Diagram

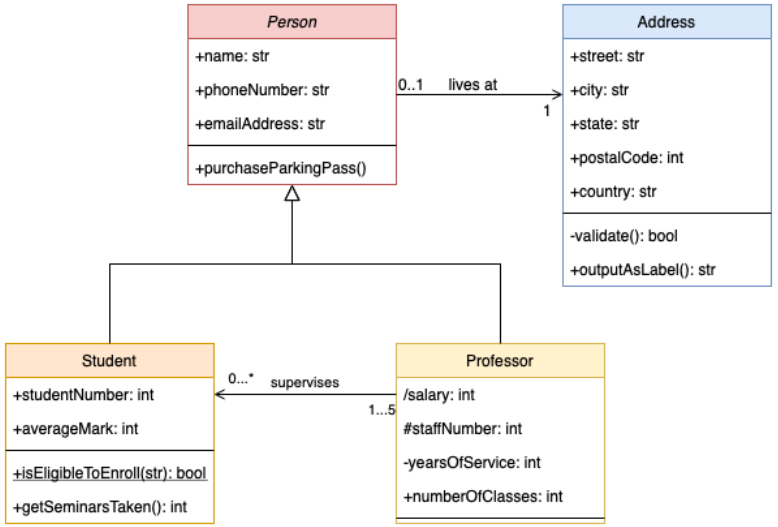


source: https://boardmix.com/tips/uml-activity-diagram/

Project Setup

The following four tasks should be considered:

- UML (Unified Modeling Language)
  - Class Diagram



source: https://www.drawio.com/blog/uml-class-diagrams

Project Setup

The following four tasks should be considered:

- Pseudo code

  Pseudo code in data science or web development is a technique used to describe the distinct steps of an algorithm in a way that’s easy for anyone with basic programming knowledge to understand.

Project Setup

The following four tasks should be considered:

- Pseudo code

- Change a numeric grade to a letter grade using the following rules:

  Grade A: score ≥ 90

  Grade B: 90 > score ≥ 80

  Grade C: otherwise

Algorithm Grade

Input: a numeric score S
Output: a letter grade

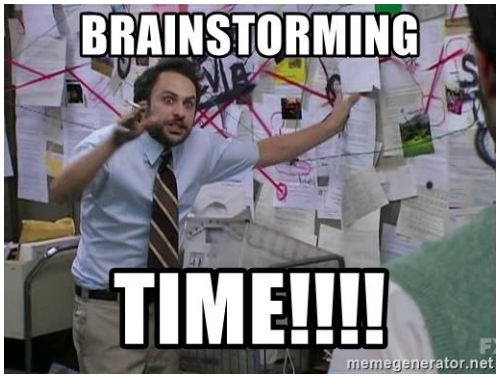
1. **If** S ≥ 90 **then**
2. **Return** grade A
3. **Endif**
4. **If** S ≥ 80 and S < 90 **then**
5. **Return** grade B
6. **Else**
7. **Return** grade C
8. **Endif**

Source: https://medium.com/@sivarasanithushna/pseudocode-and-flowchart-9c091b09b729

Project Setup

- You will be working in groups of four to solve the following four problems

- There are multiple ways to solve every problem
  - Do the necessary research
  - Outline every solution approach
  - Choose the one that is most interesting or the fastest
  - Never choose the Brute Force way



- Presentation:
  - 5 ± 2 minutes per problem
  - Explain why you chose to solve a problem a particular way
    - Do not explain the “how” (which would be the code)
  - Your fellow students should understand the problem and your approach conceptually

Exercise II

In the second exercise you will:
- Get to study multiple algorithmic problems conceptually
- Research potential solutions and describe them
- Create UML Diagrams to plan out your implementation
- Implement a chosen algorithm

Presentation:
- Record your presentation and upload it to Ilias.

If you need any help:
- See Ilias Q&A Forum

Submission deadline:
- See Ilias (roughly 3 weeks)