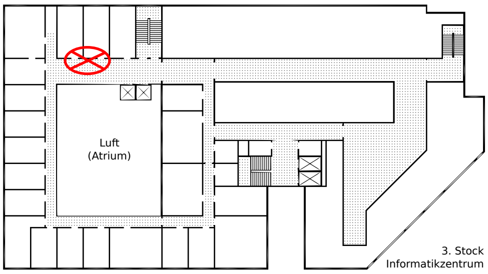
Algorithms Group **Summer '14**
Departement of Computer Science - IBR
TU Braunschweig

Michael Hemmer

Geometric Algorithms
Exercise 3
May 28, 2014



This sheet is comprised of several exercise. Most of them are theoretical but some are also applied, i.e., coding tasks. The solutions should be handed in by **June 3**. This can be done either at the beginning of the lecture or by placing them in the appropriate box of the exercise locker, see floor plan on the right. Applied exercises should be handed in via email to `hemmer@ibr.cs.tu-bs.de`. Please mark those exercises that you would like to present during the tutorial.
In order to achieve the *"Studienleistung"*, you must have presented at least three exercises until the end of the term. Moreover, you must pass the midterm exam at the beginning of June.

Exercise T1 (Planar Subdivisions - Sweep Line):

The intersection detection problem for a set S of n line segments is to determine whether there exists a pair of segments in S that intersect, i.e., finding one intersection is enough. Give a sweep line algorithm that solves the intersection detection problem in $O(n \log n)$ time.

Exercise P1 (Planar Subdivisions - Sweep Line):

Implement the algorithm above.

Exercise T2 (Expected Value):

Change the algorithm `FindIntersections` such that the working storage is $O(n)$ instead of $O(n+k)$, where k is the number of intersections. The working storage is the space that is actually required by the algorithm, i.e, already reported intersections that the algorithm does not explicitly keep in a data structure are not counted.

Exercise T3 (DCEL):

Give an algorithm that lists all neighbors of a given vertex v.

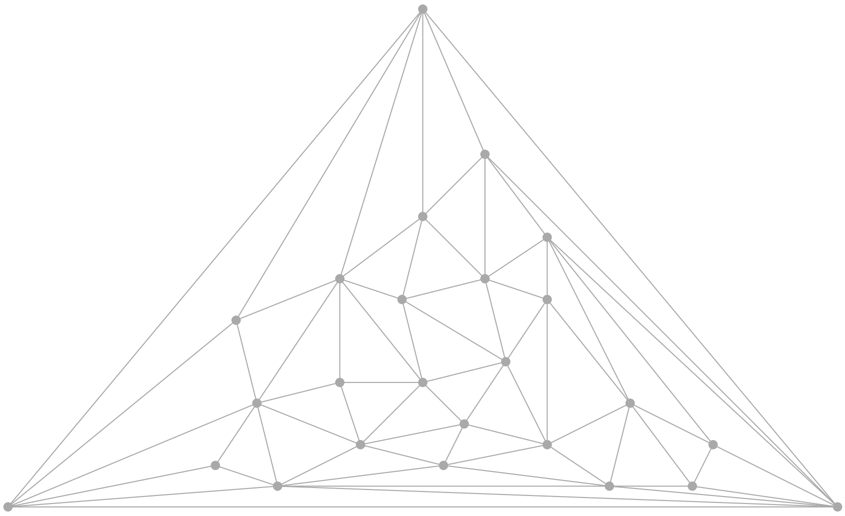
Exercise T4 (DCEL):

Give an example of a DCEL where *e.face() == e.twin().face()* for some edge e.

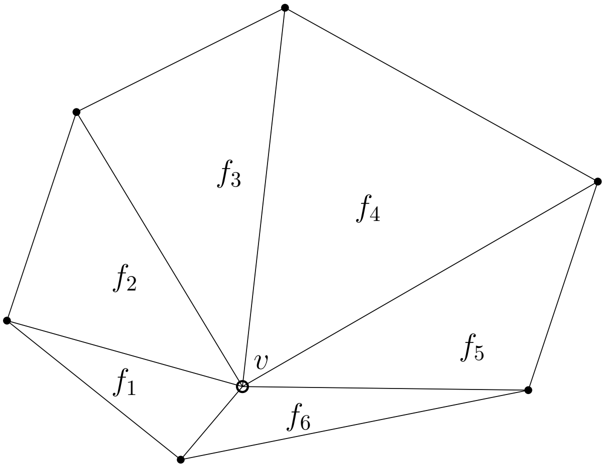
Exercise T5 (Batched Point Location - Sweep Line):

Given the DCEL of a planar subdivision S and a set of points P. Give an algorithm that locates every point of P within S, i.e., for every point $p \in P$ report p and the feature that p is located in. The algorithm should run in $O((n+m)\log(n+m))$ time, where n is the complexity of S and m is the cardinality of P.

Exercise T6 (Point Location - Kirkpatrick):



a) Given the triangulation (level 0) above. Identify a sufficiently large set of independent vertices. According to this set, give the triangulation for level 1 according to the algorithm of Kirkpatrick.



b) Given the fan of the vertex v above. Give a triangulation after v is removed and the parent/child relation according to algorithm of Kirkpatrick for the new/old faces.