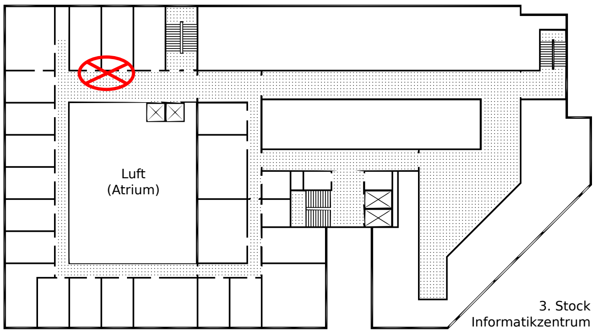
Abteilung Algorithmik **Winter 2018/2019**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Dr. Christian Scheffer
Andreas Haas

Algorithmische Geometrie
Übung 3 vom 30. 11. 2018

Abgabe der Lösungen bis zum Freitag, den 14. 12. 2018 um 11:30 im Hausaufgabenrückgabeschrank.



Bitte die Blätter vorne deutlich mit eigenem Namen sowie Matrikel- und Gruppennummer versehen!

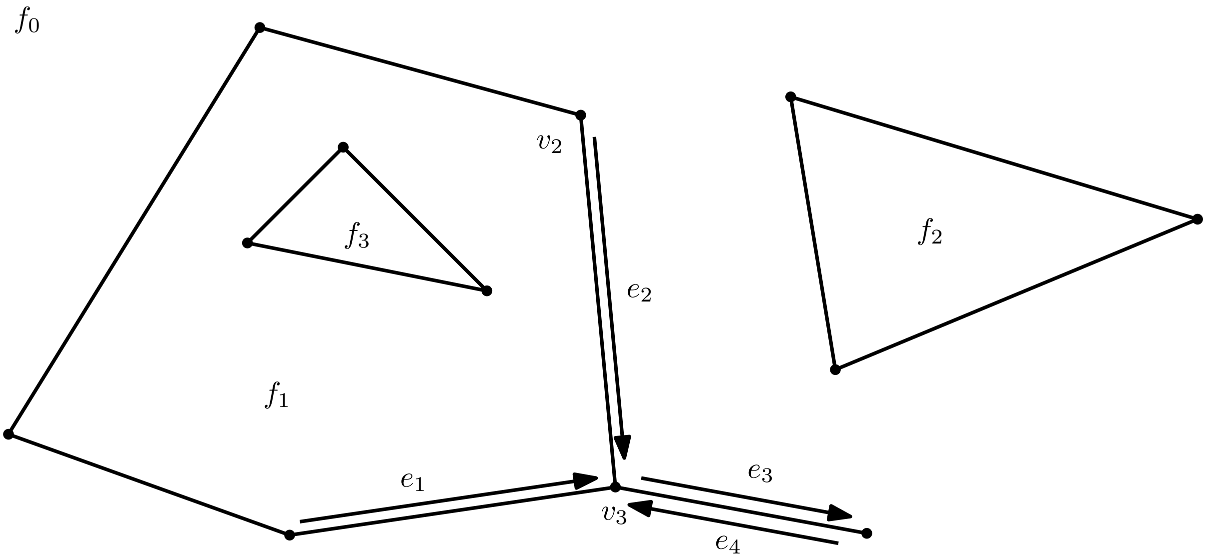


Abbildung 1: DCEL

Aufgabe 1 (DCEL):

a) Betrachte Abbildung 1, gib jeweils eine Sequenz von Aufrufen an um
- v_3 von e_1
- e_4 von e_1
- f_1 von e_4
- f_1 von f_3
- f_0 von e_1

aus zu erreichen.

Beispielsweise ist v_3 von e_2 erreichbar durch:

$$v_3 = e_2.next().source()$$

b) Gib einen Algorithmus an, der alle Nachbarn eines Knotens v aufzählt bzw. durchläuft.

(5 Punkte)

Aufgabe 2 (Eulers Formel): Sei $G = (V, E)$ ein einfacher, planarer, zusammenhänger Graph. Zeige, dass für eine kreuzungsfreie Einbettung von G in der Ebene folgendes gilt:

a) Sei F die Anzahl Flächen (die unbegrenzte Fläche zählt dazu), dann gilt

$$|V| - |E| + |F| = 2$$

b) Für $|V| \geq 3$ gilt,

$$|E| \leq 3|V| - 6$$

c) Der durchschnittliche Knotengrad ist echt kleiner als 6, d.h.

$$\frac{1}{|V|} \sum_{v \in V} \delta(v) < 6$$

(5 Punkte)

Aufgabe 3 (Batched Point Location): Sei die DCEL einer ebenen Unterteilung S und eine Menge P von Punkten gegeben. Gib einen Algorithmus an, der jeden Punkt $p \in P$ innerhalb von S lokalisiert, d.h. der Algorithmus soll für jeden Punkt die Fläche in der er sich befindet bzw. die Kante oder den Knoten auf dem er sich befindet ausgeben. Die Worst-Case Laufzeit soll dabei in $O((n + m)\log(n + m))$ liegen, wobei n die Anzahl der Segmente in S und m die Anzahl der Punkte in P ist.

Begründe wieso dein Algorithmus die verlangte Laufzeit erfüllt.

Bemerkung: Der Aufbau einer Trapezoidal Map hat lediglich eine erwartete Laufzeit von $O(n \log n)$ und eine Abfrage darin lediglich eine erwartete Laufzeit von $O(\log n)$.

(5 Punkte)

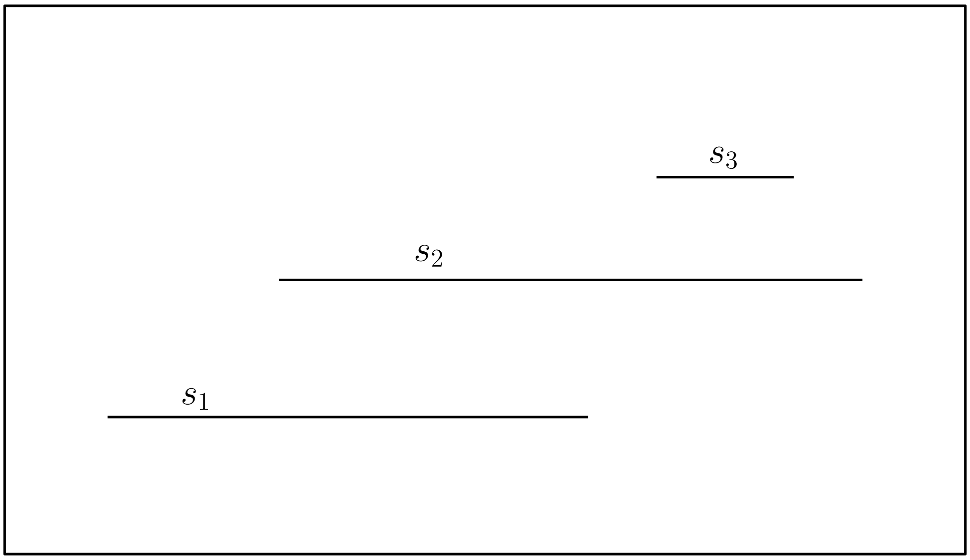
Aufgabe 4 (Planar Point Location - Trapezoidal Map): Füge die Segmente (s_1, s_2, s_3) entsprechend ihrer Reihenfolge ein.

a) Gib $\mathcal{T}(S_i)$ und $D(S_i)$ für $i \in \{1, 2, 3\}$ separat an.

Beachte, dass nicht alle Pfade in $D(S_3)$ valide Suchpfade sind. Sei $\mathcal{D}$ die maximale Länge aller Pfade in $D(S)$ und $\mathcal{L}$ die maximale Länger aller validen Suchpfade.

b) Markiere einen Pfad der Länge $\mathcal{D}$ in $D(S_3)$.

c) Markiere einen validen Suchpfad der Länge $\mathcal{L}$ in $D(S_3)$.



Bonus (SCHWIERIG):

- Sei n eine Quadratzahl, d.h., $n = k^2$ für ein $k \in \mathbb{N}$. Skizziere eine Menge von n Segmenten S und gib eine Einfügereihenfolge an, sodass die Länge des längstes Pfades in $D(S)$ in $\Omega(n)$ liegt, aber die Länge des längsten gültigen Suchpfades in $O(\sqrt{n})$.
- Sei n eine Zweierpotenz, d.h., $n = 2^k$ für ein $k \in \mathbb{N}$. Skizziere eine Menge von n Segmenten S und gib eine Einfügereihenfolge an, sodass die Länge des längstes Pfades in $D(S)$ in $\Omega(n)$ liegt, aber die Länge des längsten gültigen Suchpfades in $O(\log n)$.

(5 + 5 Bonus Punkte)