Tafelbild

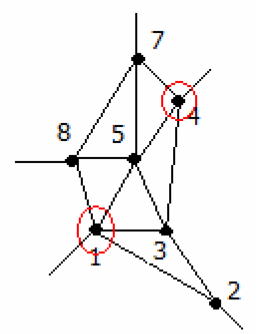


Bild 1

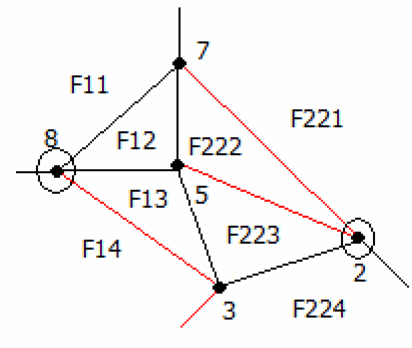


Bild 2

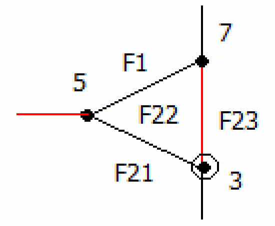


Bild 3

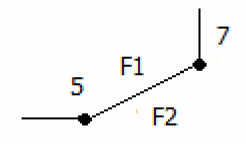


Bild 4

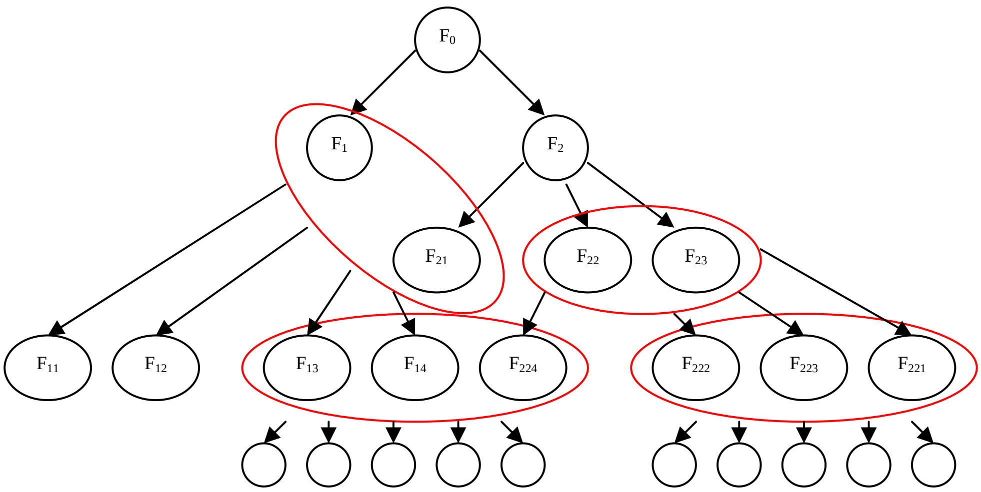


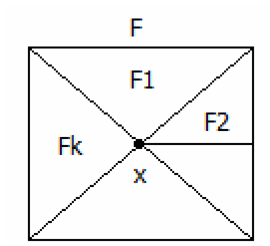
Bild 5

Algorithmus:

(1) Bestimme unabhängige Knotenmenge.
(2) Elemente von I entfernen aus dem geometrischen Graphen.

Im Beispiel: I = {1, 4}. Die nehmen wir heraus. Bleibt schwarzer Graph (Bild 2). Was entsteht, muss nicht triangulierter Graph sein, also:

(3) Trianguliere „Löcher“.
(4) Rekursion (wir bauen rekursiv die Datenstruktur).
(5) Wir fassen entsprechende Blätter zusammen.
(6) Wurde Element x im Schritt (2) entfernt, dann wird es jetzt eingefügt.

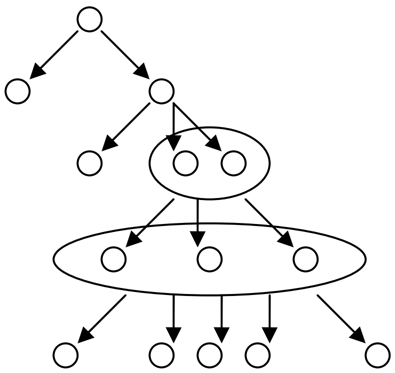


I = {1, 4} (Bild 1)
I = {2, 8} (Bild 2)
I = {3} (Bild 3)

Blätter = Dreiecke der entsprechenden Unterteilung (Bild 5)

<u>Suchen in P</u> <u>(Höhe des „Baumes“)</u>

Wie groß ist die Höhe?
$h \approx \log_{24/23} n \approx 16,29 \log_2 n$



$(23/24)^k n \leq 1$

$\leq 23/24\, n$

$\leftarrow n$

In der Praxis ist der Faktor 16, 29 nicht zu vernachlässigen. Man kann das aber verbessern.

<u>Speicherplatz</u> <u>(d.h. Platz, den man braucht, um diese Struktur zu speichern)</u>

$$\Theta(\underset{\uparrow}{n} + 23/24n + (23/24)^2 n + \dots) = \Theta(n)$$

für die unterste Ebene

weil: $\leq n \sum_{i=0}^{\infty} (23/24)^i$

------------- konv. Reihe = 1/(1 – 23/24) = 24

<u>Vorverarbeitungszeit</u> <u>(d.h. die Zeit zur Konstruktion der Datenstruktur)</u>

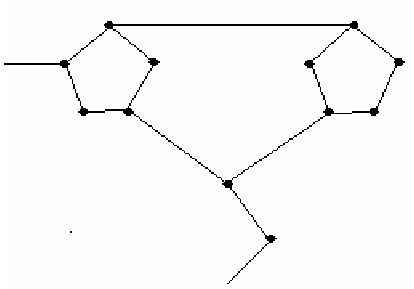
Schritt (1):	O(n) (siehe Lemma)
Schritt (2):	(Elemente aus I entfernen, und ausschließlich die Kanten) O(n) (weil das ein planarer Graph ist)
Schritt (3):	(jedes Loch ist ein einfacher Polygon) [Hier haben Löcher konstante Größe und daher konstante Zeit] O(n) Löcher konstanter Größe, O(1) Zeit pro Loch, also insgesamt: O(n)
Schritt (4):	(Rekursion) T(dn), wobei d = 1 – c = 23/24
Schritt (5):	(Wir fassen entsprechende Blätter zusammen) O(n)

Schritt (6): (Praktisch der Schritt (2) rückgängig: für jede entstandene Facette konstante Zeit) $O(n)$

Rekursionsgleichung (Schritt 4):

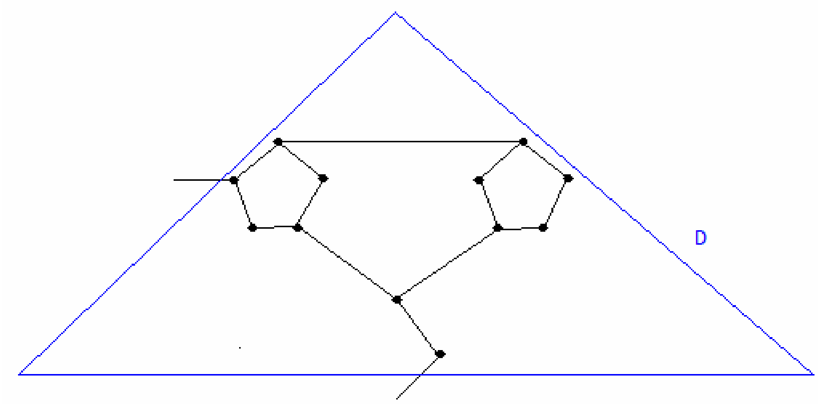
$$
\begin{aligned}
T(2) &= \text{const} \\
T(n) &= T(dn) + O(n) \\
&= T(dn) + an \qquad\qquad a = \text{Konstante} \\
&\;. \\
&= an + adn + ad^2n + \dots \\
&= an(\underbrace{1 + d + d^2 + \dots}_{\leq 1/(1 - d) \text{ konv. Reihe}}) \\
&= O(n)
\end{aligned}
$$

Verallgemeinerung auf beliebige ebene Unterteilungen:

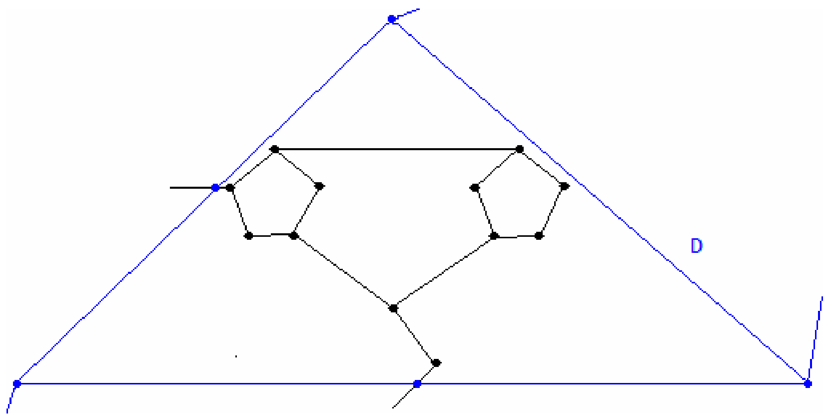


Wie kann man den Algorithmus in diesem Fall anwenden?

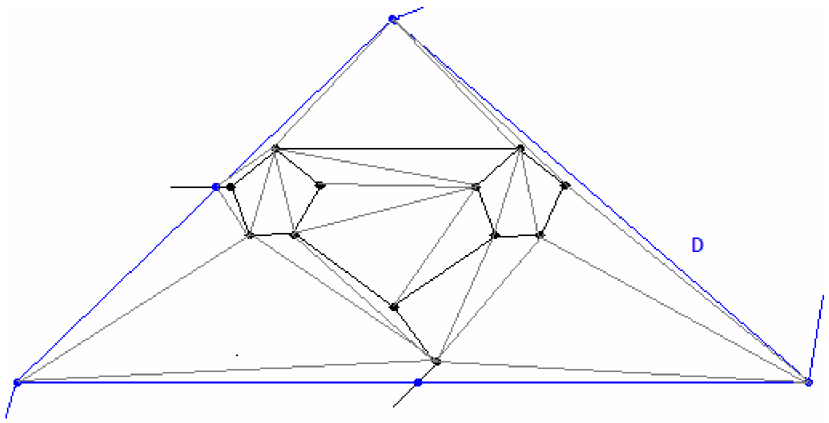
(1) Wir legen zuerst ein „großes“ Dreieck D in die Ebene, das den endlichen Teil der Unterteilung umschließt (d.h. alle Strecken).



Wir lassen von Ecken von D Strahlen ausgehen und nehmen Schnittpunkte von D mit den Strecken mit auf



(2) Trianguliere alle Polygone im Inneren von D.



(3) Wende die vorherige Methode an.

<u>Zeit:</u> (1) O(n)
(2) Triangulierung einfacher Polygone geht in O(n) Zeit (Chazelle ´91)
(3) O(n)

Satz: Zu einer ebenen Unterteilung der Größe n kann in O(n) Verarbeitungszeit eine Datenstruktur mit Platzbedarf O(n) konstruiert werden, die jede point-Location-Anfrage in O(log n) Zeit beantworten kann.

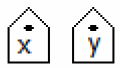
Anwendung: Das Post-Office-Problem ist lösbar mit Vorverarbeitungszeit O(n log n), Platzbedarf O(n), Suchzeit O(log n).

Denn: Konstruiere Voronoi-Diagramm – und daraus point-location-Datenstruktur.

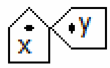
Weitere Anwendungen von Voronoi-Diagramm etc.:

Lemma:

Sei S eine endliche Menge von Punkten in R^2, $S \subset R^2$ endlich. Seien $x, y \in S$ mit $x \neq y$ und $|VR(x) \cap VR(y)| \leq 1$.

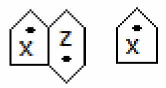


oder



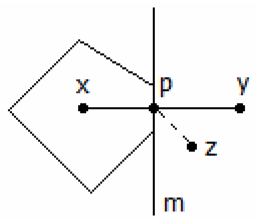
Dann :
$\exists\, z \in S$ mit $||x - z|| \leq ||x- y||$ und $||y - z|| < ||x - y||$ und VR(x), VR(z) haben eine gemeinsame

begrenzende Kante (zu a.W. $\overline{xz}$ liegt in der Delamay-Triangulierung).



Beweis:

Betrachte Strecke s = xy.
p sei Schnittpunkt von s mit Rand von VR(x).



z ($\neq$ y), so dass (nicht entartete) Kante zwischen VR(x), VR(z) von s geschnitten wird.

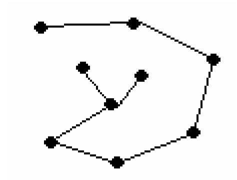
$||y - z|| \leq ||x - y||$ weil y, z beide auf derselben Seite von m liegen
$y \neq p$ denn es kann nicht auf V-Kante liegen

also:
$||y - z|| < ||x - y||$
$\uparrow$ weil nur p die Eigenschaft hat, gleichen Abstand zu x und z zu haben
$y \neq p$
und $||x - p|| \leq ||y - p||$ da $y \notin$ Inneren des Kreises um p mit dem Radius $||x - p||$

$||x - y|| = ||x - p|| + ||y - p||$ weil p auf der Strecke von x nach y liegt
$||x - y|| = ||x - p|| + ||y - p|| > 2||x - p|| = ||x - p|| + ||z - p|| \geq ||x - z||$

□

Euklidischer minimal spannender Baum, EMSB



Gegeben: $S \subset R^2$ endlich
Finde: Baum mit der Knotenmenge S, geradlinigen Kanten, so dass Summe der Kantenlängen minimal ist

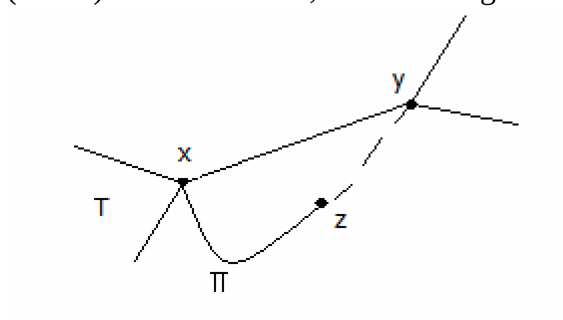
Behauptung:
Es gibt zumindest einen EMSB, dessen Kanten alle zur Dalamay-„Triangulierung“ von S gehören.

Beweis zur Behauptung:
Sei T ein beliebiger EMSB mit einer Kante $\overline{xy}$, die nicht zur D.-T. gehört.

$\Rightarrow \ \exists$ z mit $\|x - z\| \leq \|x - y\|$

Lemma $\qquad \|y - z\| \leq \|x - y\|$

(oBdA) Nehmen wir an, dass ein Weg Π von x nach z existiert, der y nicht enthält.



Dann ersetzen wir die Kante $\overline{xy}$ durch die Kante $\overline{yz}$.
Das erhöht die Kosten nicht und es bleibt ein Baum - auch ein EMSB.

So können wir jede Kante ersetzen.

Daraus folgt:

Satz: Ein EMSB einer Menge S kann in Zeit O(n log n) berechnet werden, n = |S|.

Beweis: Erst Dalamay, dann Kruskal.