

Algorithmische Geometrie, Sommersemester 2025

erwartet



Trapezzerlegung eines einfachen Polygons in $O(n \log^* n)$ Zeit

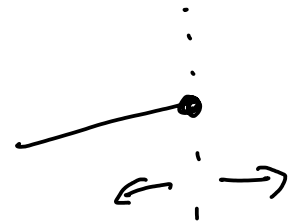
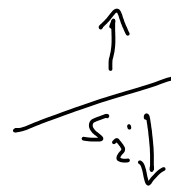
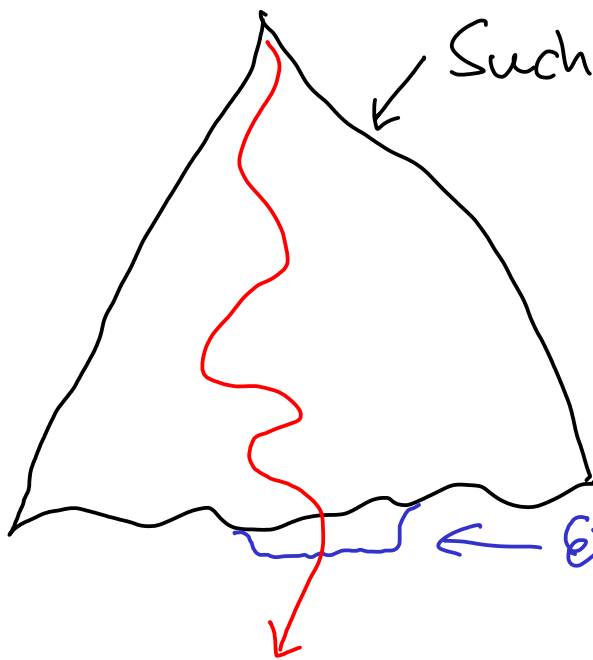
↳ Triangulierung

$$= \min \{i \mid \underbrace{\log_2 \log_2 \log_2 \dots \log_2 n}_{i\text{-mal}} \leq 1\}$$

Raimund Seidel (1991)

Bernard Chazelle (1990): $O(n)$

Randomisierte inkrementelle Konstruktion.



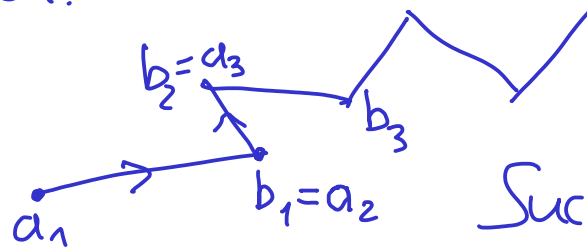
← Einfügen einer neuen Strecke
 $E(\text{Laufzeit}) = O(1)$

$\Pr[\text{Suchpfad für ein Punkt } q \text{ ist durch Einfügen von } s_i \text{ betroffen}] \leq \frac{4}{i}$

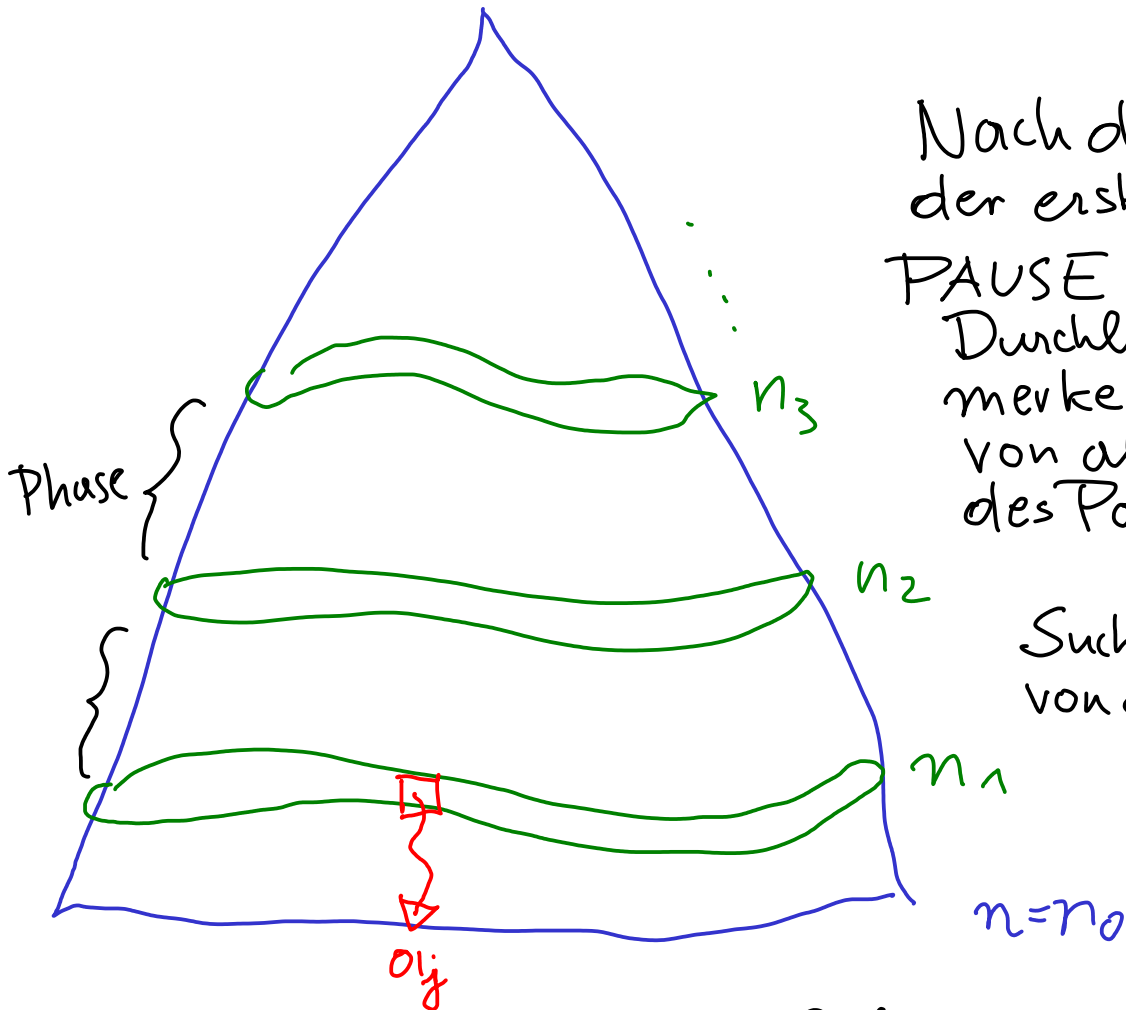
Einfügen einer neuen Strecke $s_i = a_i b_i$

- a_i suchen $O(\log i)$
- von a_i nach b_i wandern, ausbessern $O(1)$

Einfaches Polygon:



Suchzeit einsparen!



Nach dem Einfügen
der ersten n_k Strecken:
PAUSE:
Durchlaufe das Polygon,
merke die Trapeze
von allen Endpunkten
des Polygons:

Suchen nach a_j kann
von diesem Trapez aus
stehen.

S_i ... Suchstruktur nach dem Einfügen von $S_1, \dots, S_i$

Trapez von a in S_i bekannt

$$\begin{aligned} \text{Suche von } a \text{ in } S_n: & O\left(\frac{1}{i+1} + \frac{1}{i+2} + \dots + \frac{1}{n}\right) \\ & = O(\log n - \log i) = O\left(\log \frac{n}{i}\right) \end{aligned}$$

ZIEL: Einfügen aller Strecken in einer PHASE
(zwischen n_{k+1} und n_k) in $O(n)$ Zeit.

$$\# \text{ Suchen} = O(n_k)$$

$$\text{Suchzeit} = O\left(\log \frac{n}{n_{k+1}}\right)$$

$$\text{Suchzeit für Phase } (n_k - \cancel{n_{k+1}}) \cdot O\left(\log \frac{n}{n_{k+1}}\right) \stackrel{!}{=} O(n)$$

$$n_k \cdot \log \frac{n}{n_{k+1}} = n$$

$$\log \frac{n}{n_{k+1}} = \frac{n}{n_k}$$

$$2^{n/n_{k+1}} = \frac{n}{n_{k+1}}$$

$$n_k = \frac{n}{H_k}$$

$$H_k = 2^{H_{k-1}}$$

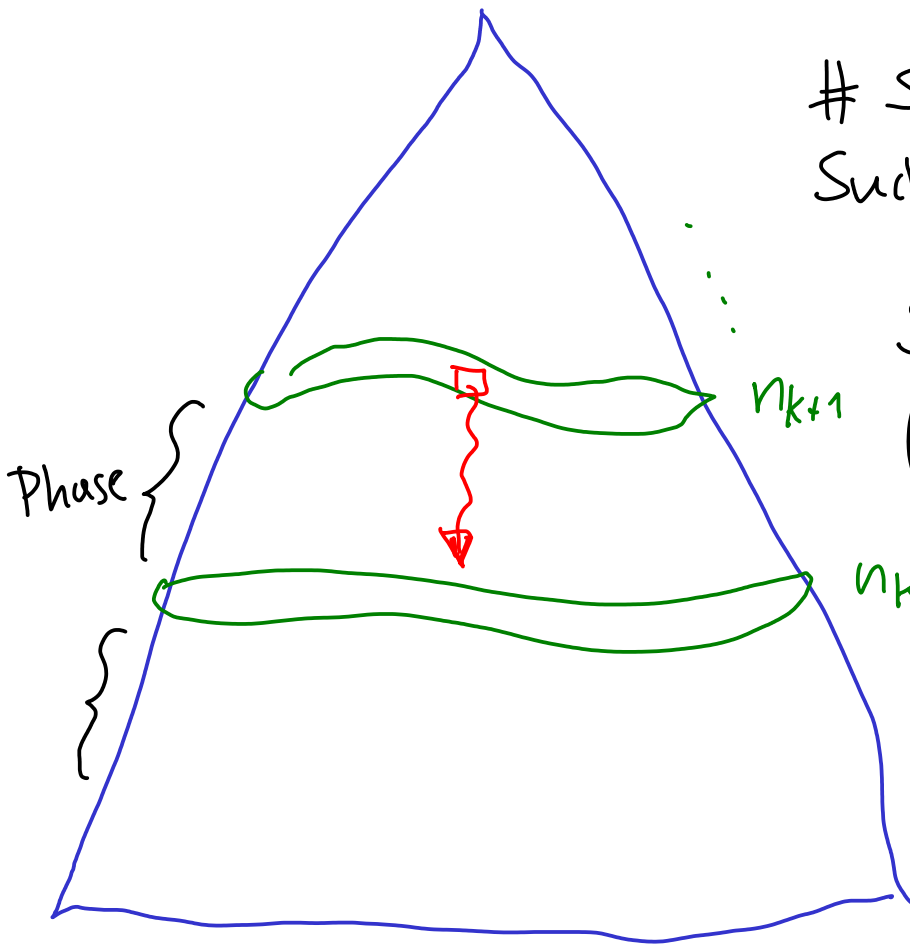
$$H_4 = 2^{2^{2^2}} = 2^{H_3}$$

$$H_3 = 2^{2^2}$$

$$n_2 = \frac{n}{H_2} \quad H_2 = 2^2$$

$$n_1 = \frac{n}{H_1} \quad H_1 = 2^1 = 2$$

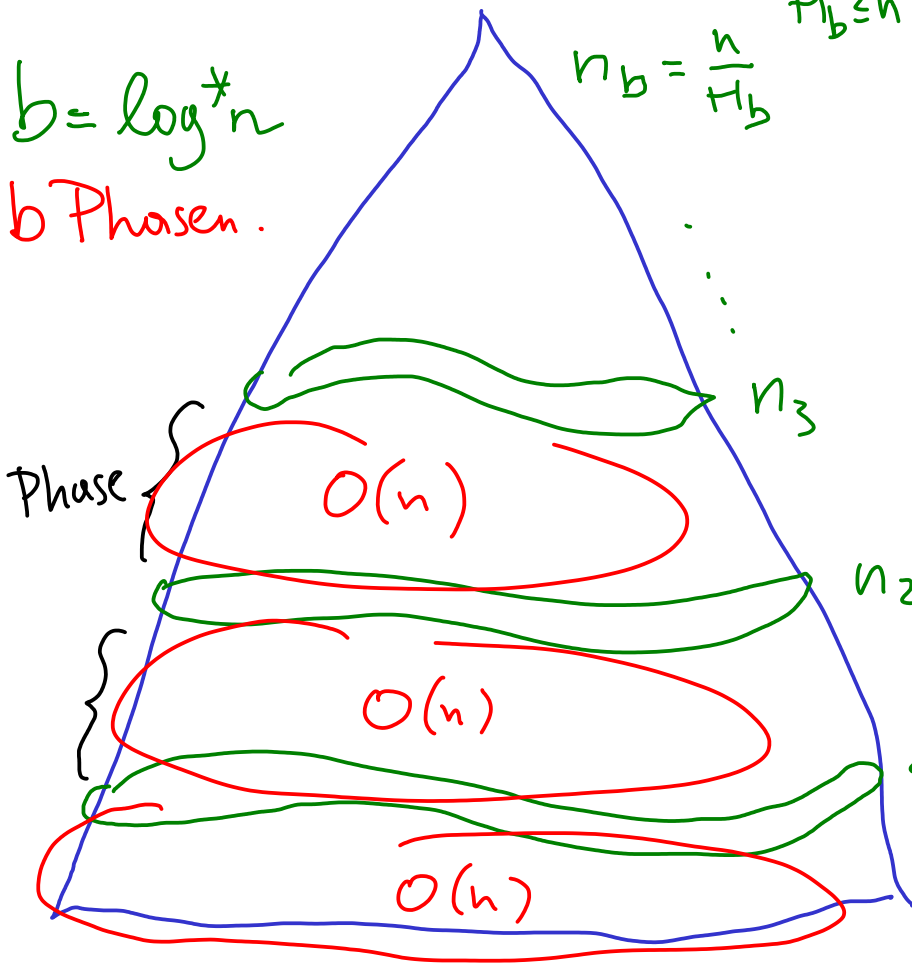
$$n = n_0 = \frac{n}{H_0} \quad H_0 = 1$$



$$b = \log^* n$$

b Phasen.

$$n_b = \frac{n}{H_b} \quad H_b \leq n$$



$$b \text{ Phasen} \times O(n) \text{ Zeit} = O(bn) = O(n \log^* n)$$

Funktioniert für jede zusammenhängende
ebene Unterteilung



Tiefensuche

→ Kantenzug mit
Wiederholungen,
besucht jede Kante 2x.