Karlsruher Institut für Technologie

Übung 9 – Algorithmen II

Daniel Seemaier, Tobias Heuer – *daniel.seemaier@kit.edu, tobias.heuer@kit.edu*

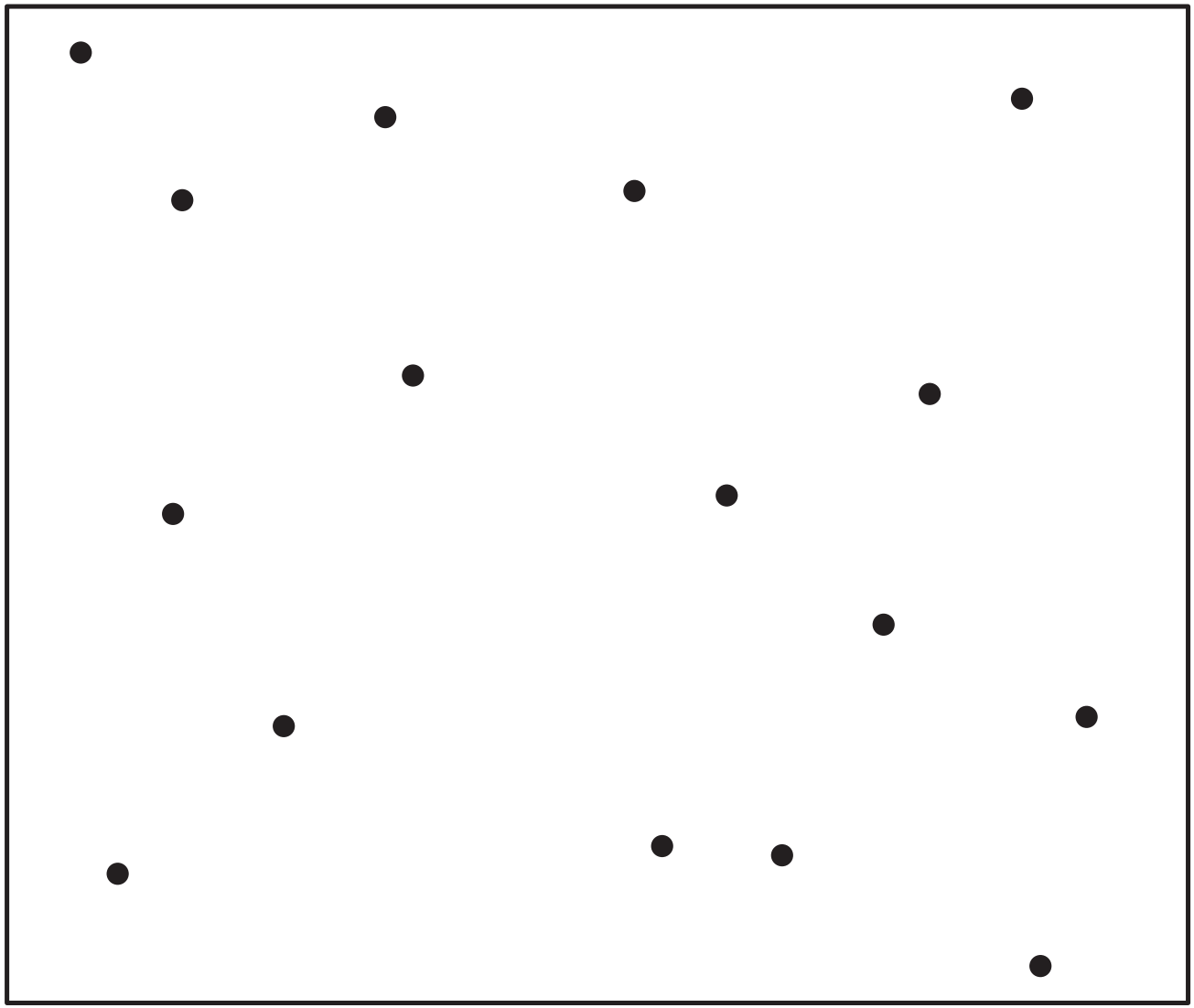
`http://algo2.iti.kit.edu/AlgorithmenII_WS20.php`

Institut für Theoretische Informatik - Algorithmik II

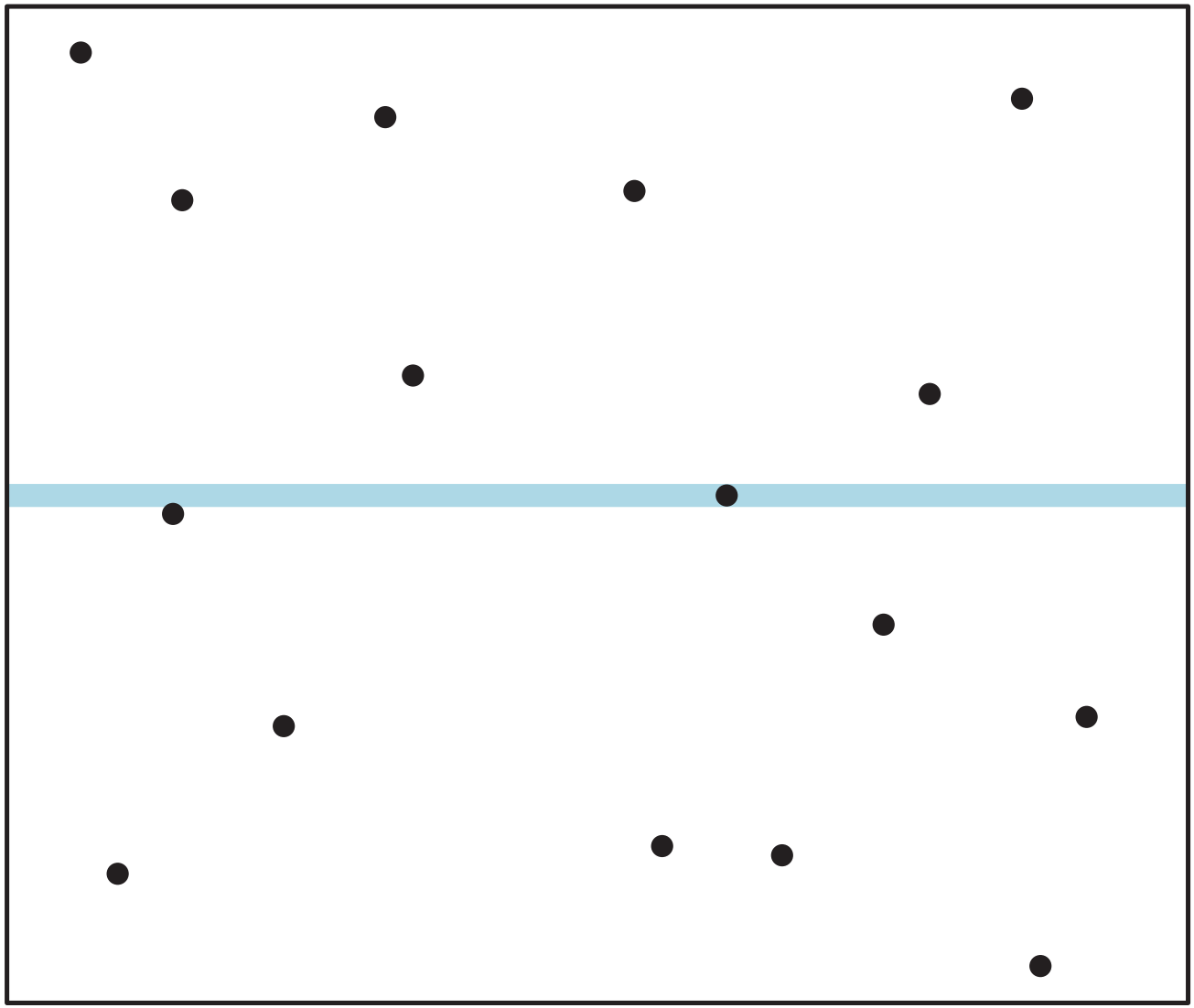
Themenübersicht

- Geometrische Datenstrukturen
  - Übersicht
  - Wavelet Tree
  - K^2-Treaps
- Konvexe Hülle
  - Punktorientierung
  - Graham-Scan Algorithmus
  - Durchmesser Punktmenge

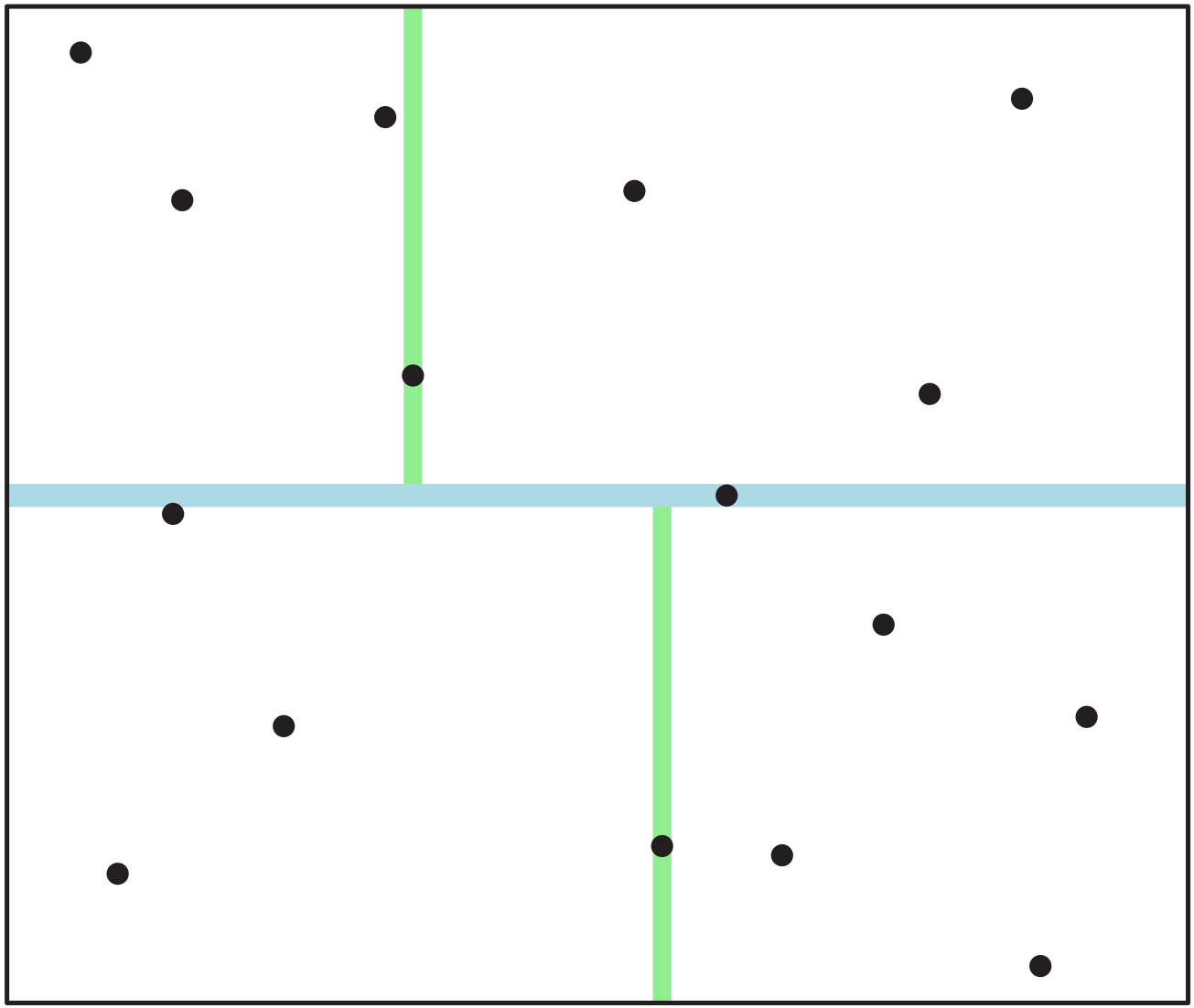
k-d-Tree



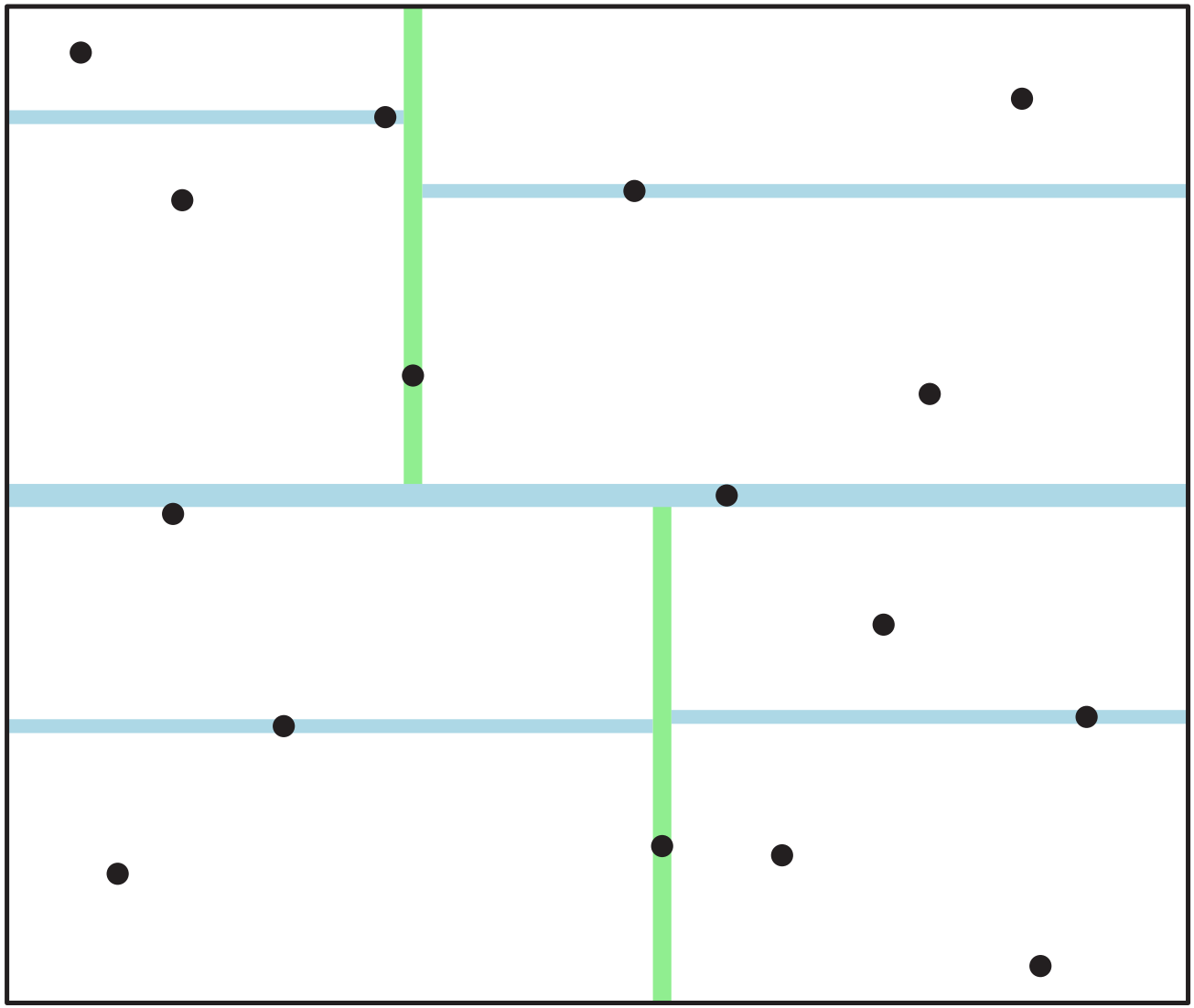
k-d-Tree



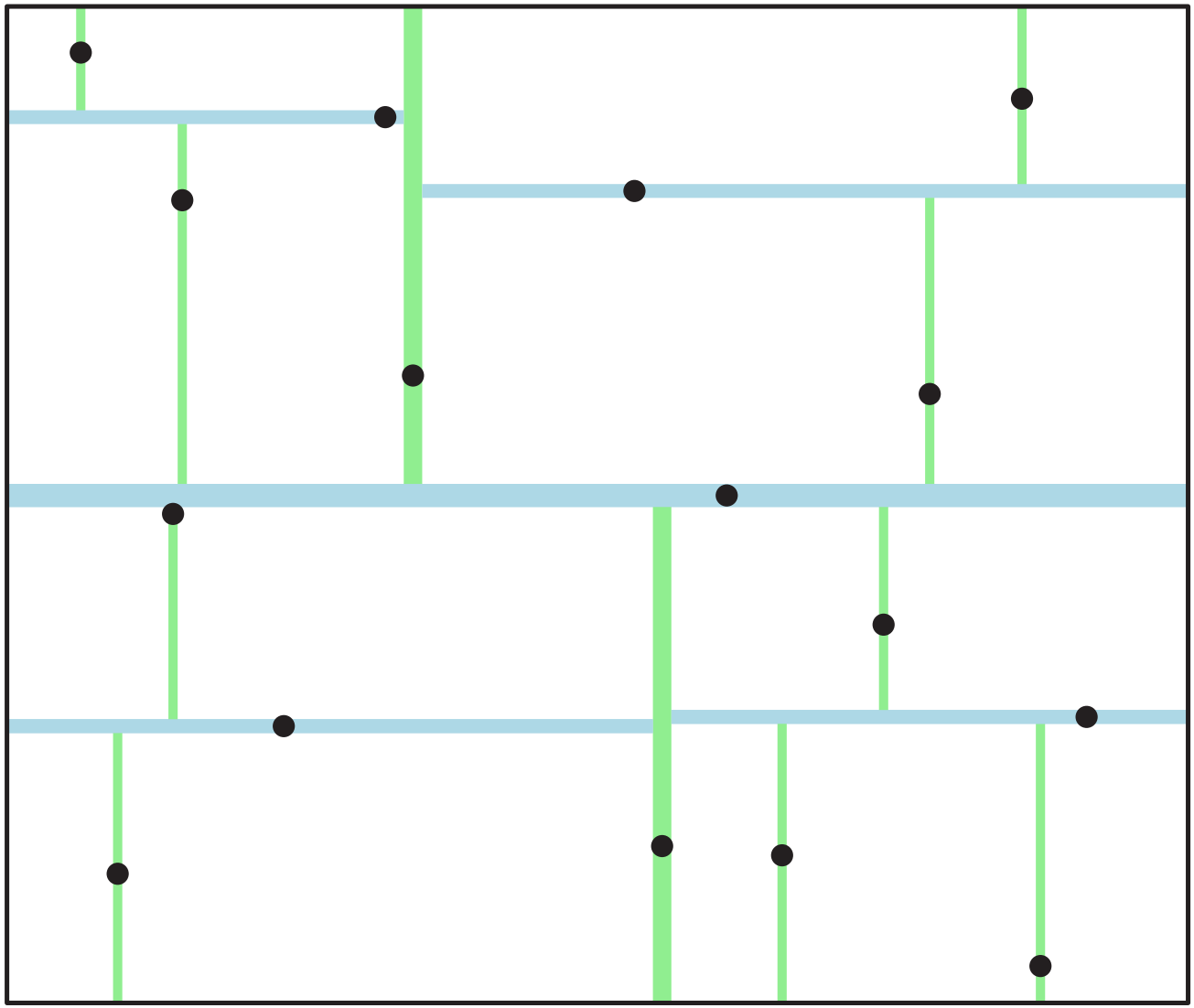
k-d-Tree



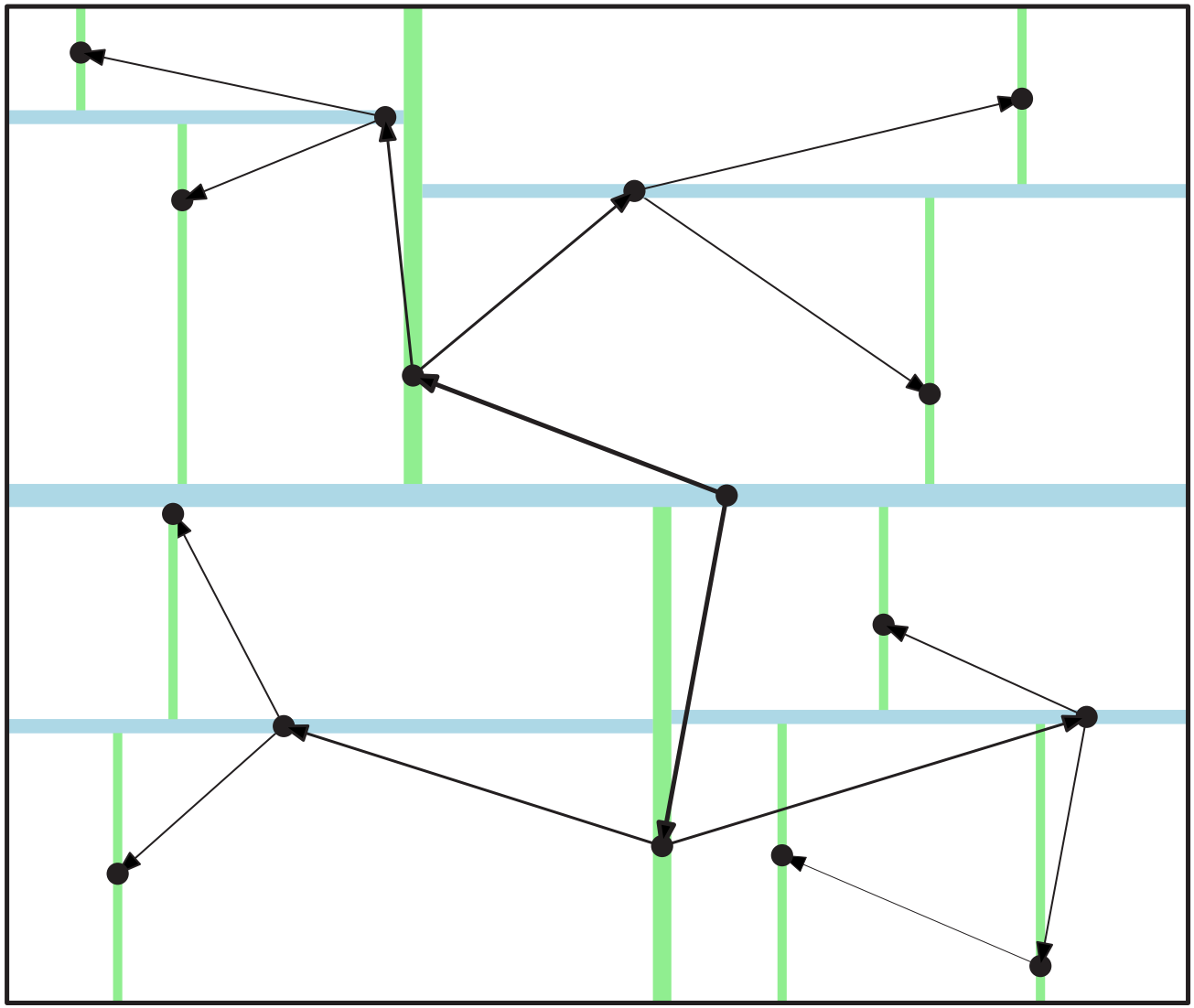
k-d-Tree



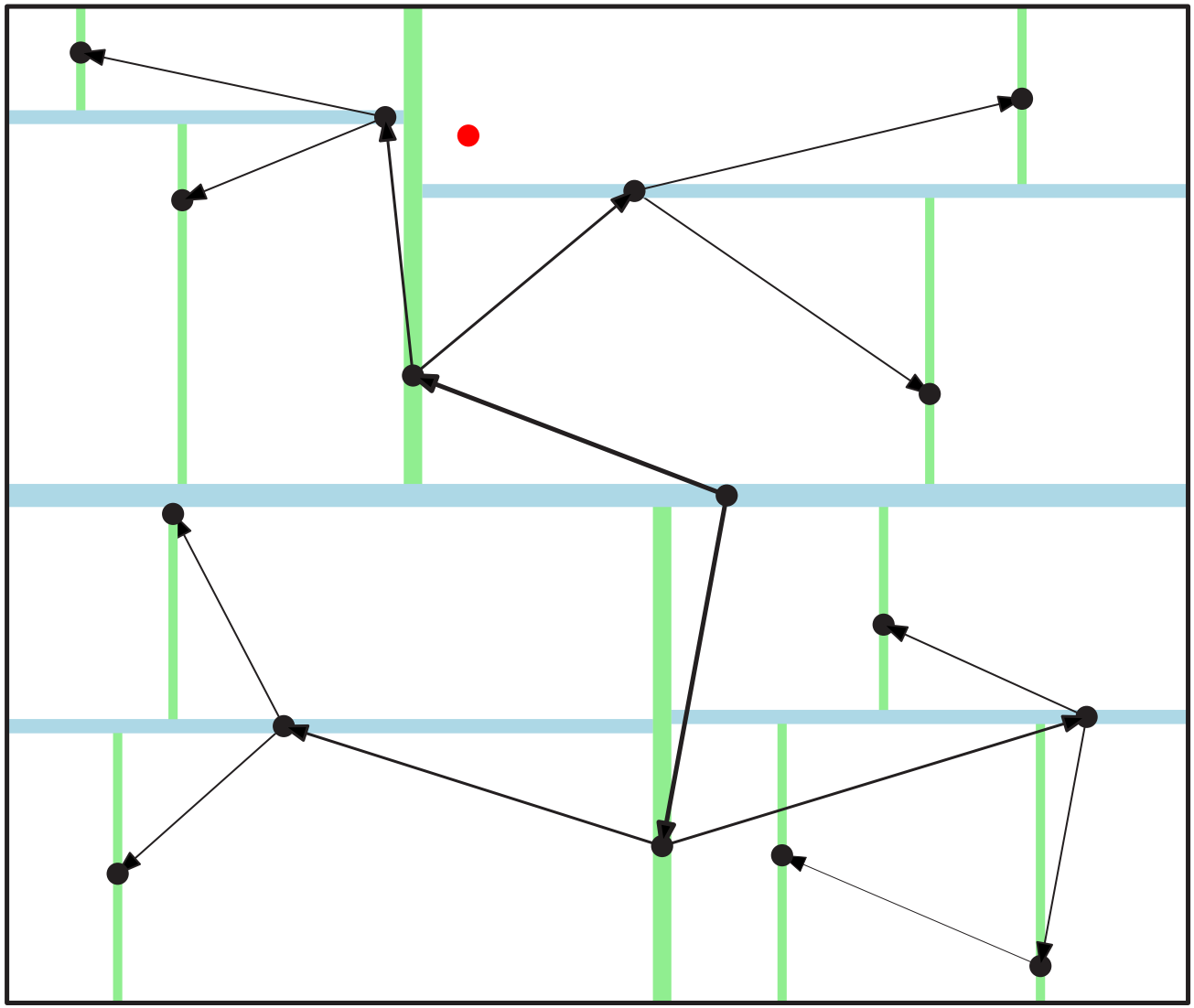
k-d-Tree



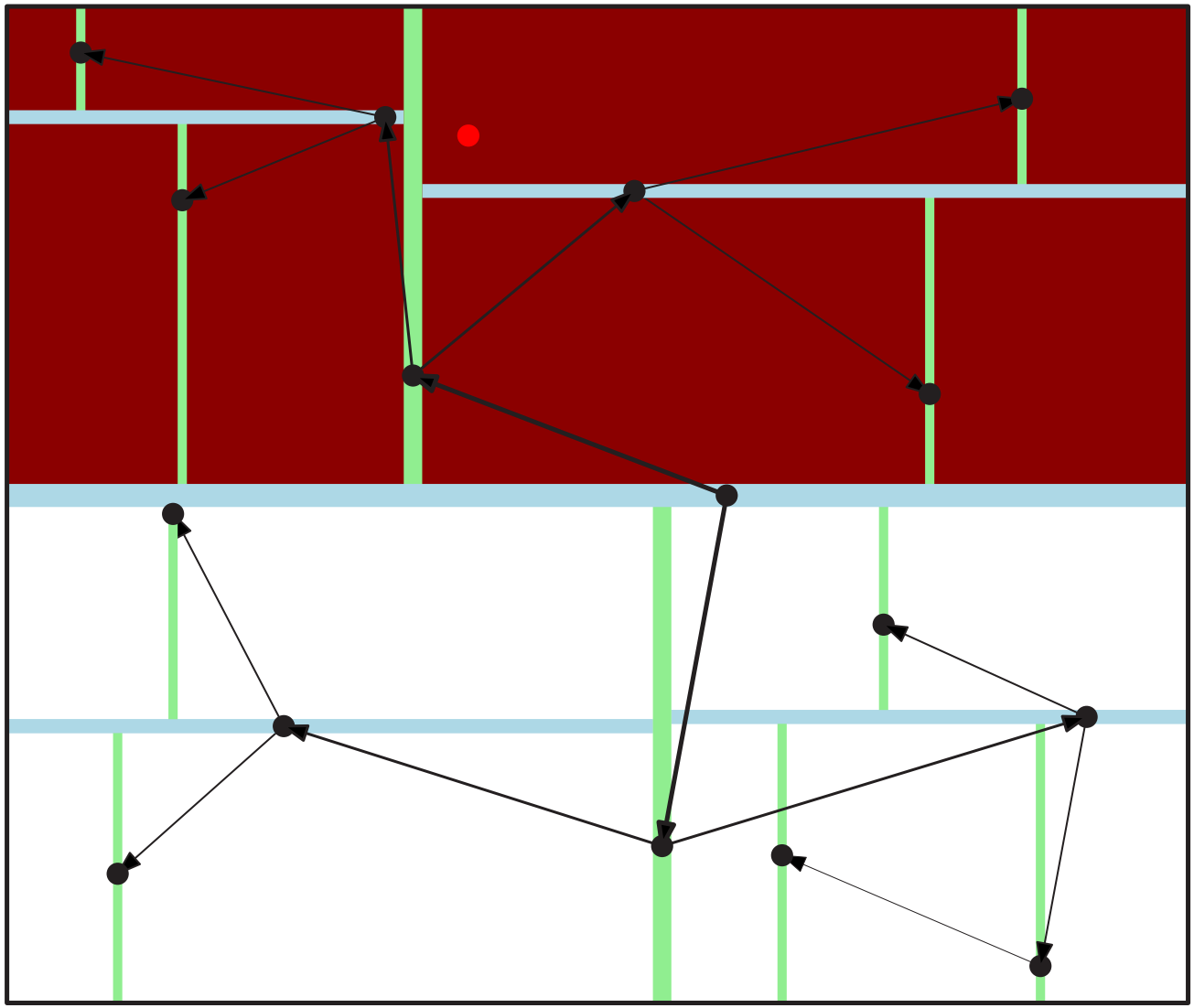
k-d-Tree



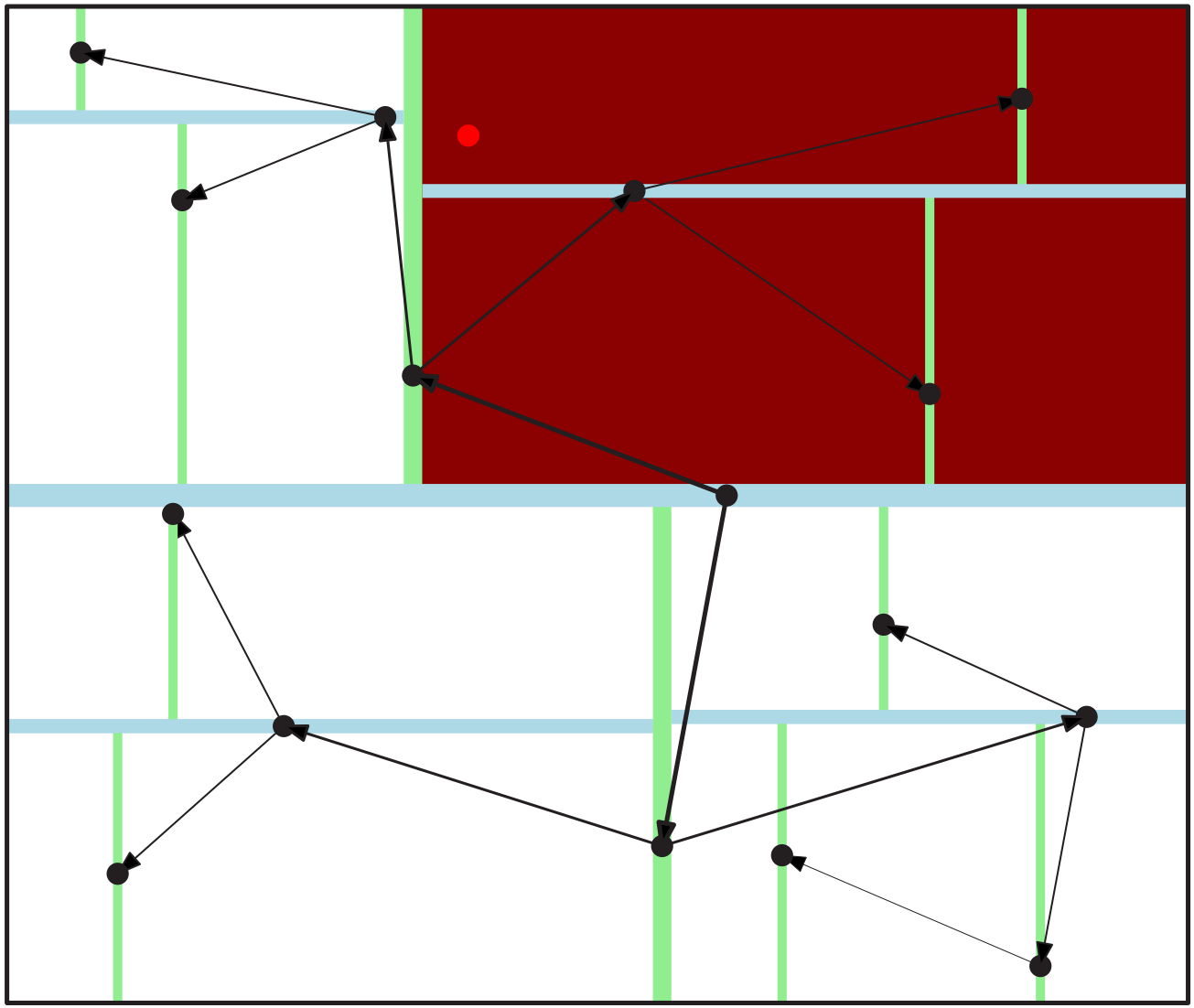
k-d-Tree



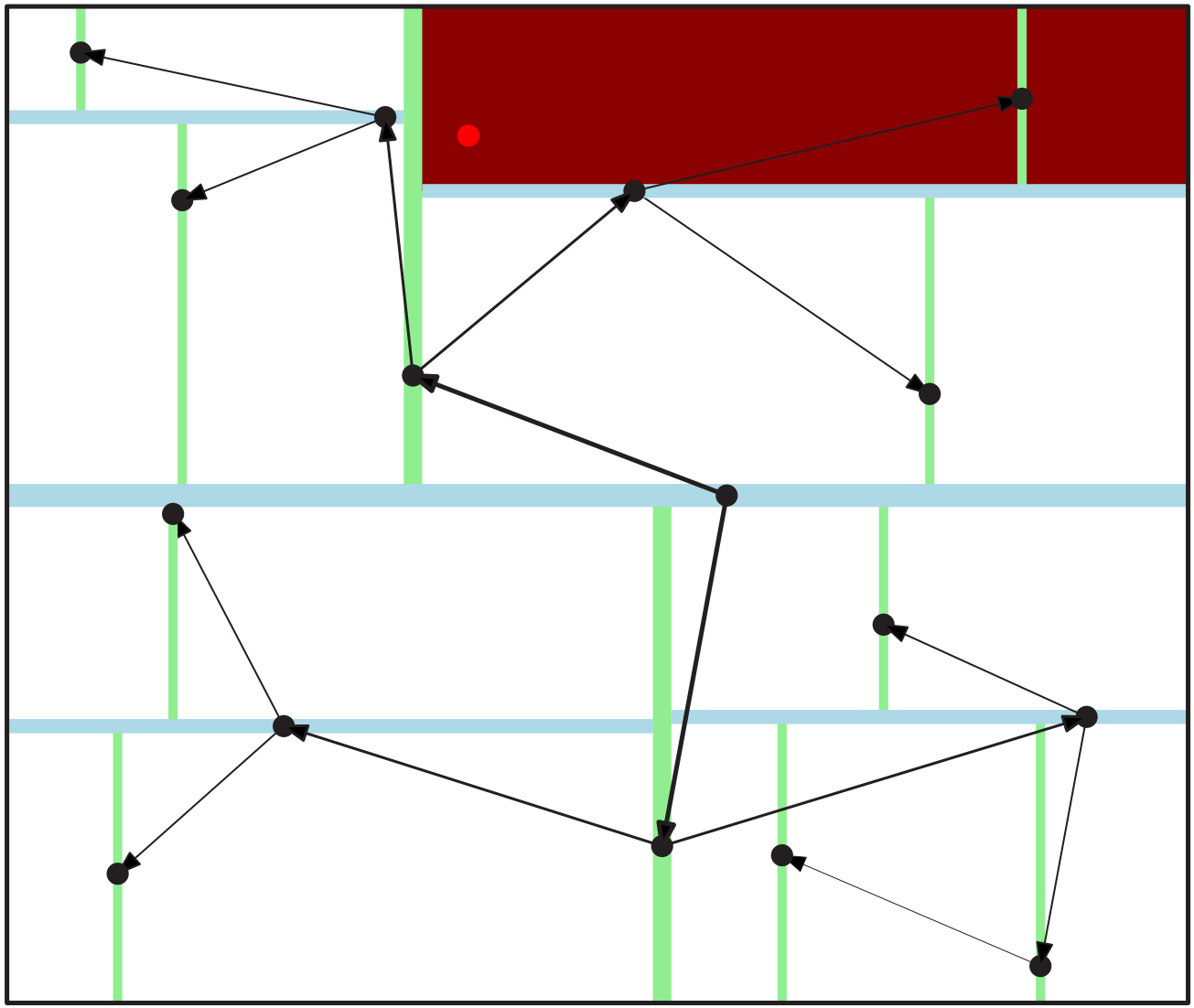
k-d-Tree



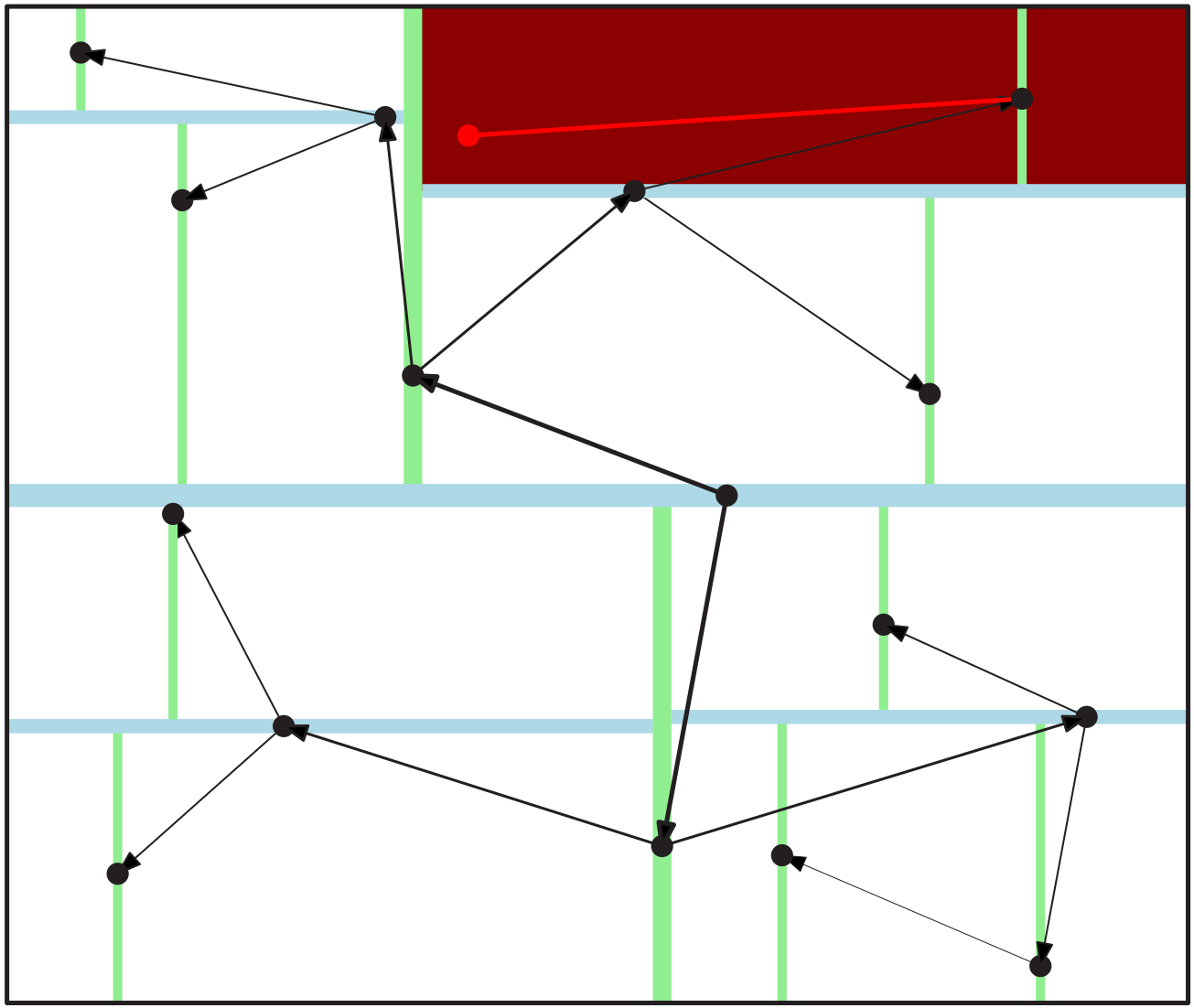
k-d-Tree



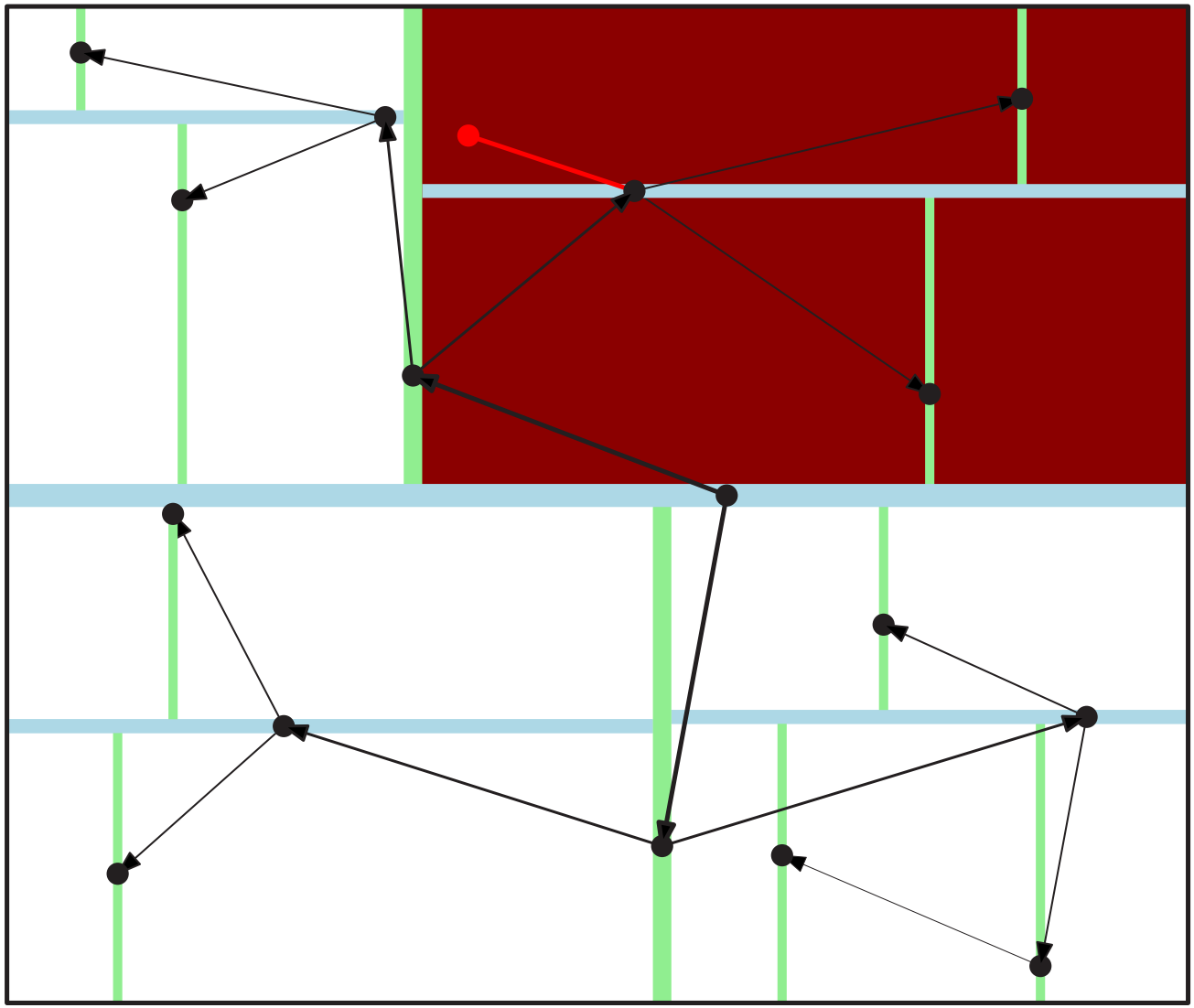
k-d-Tree



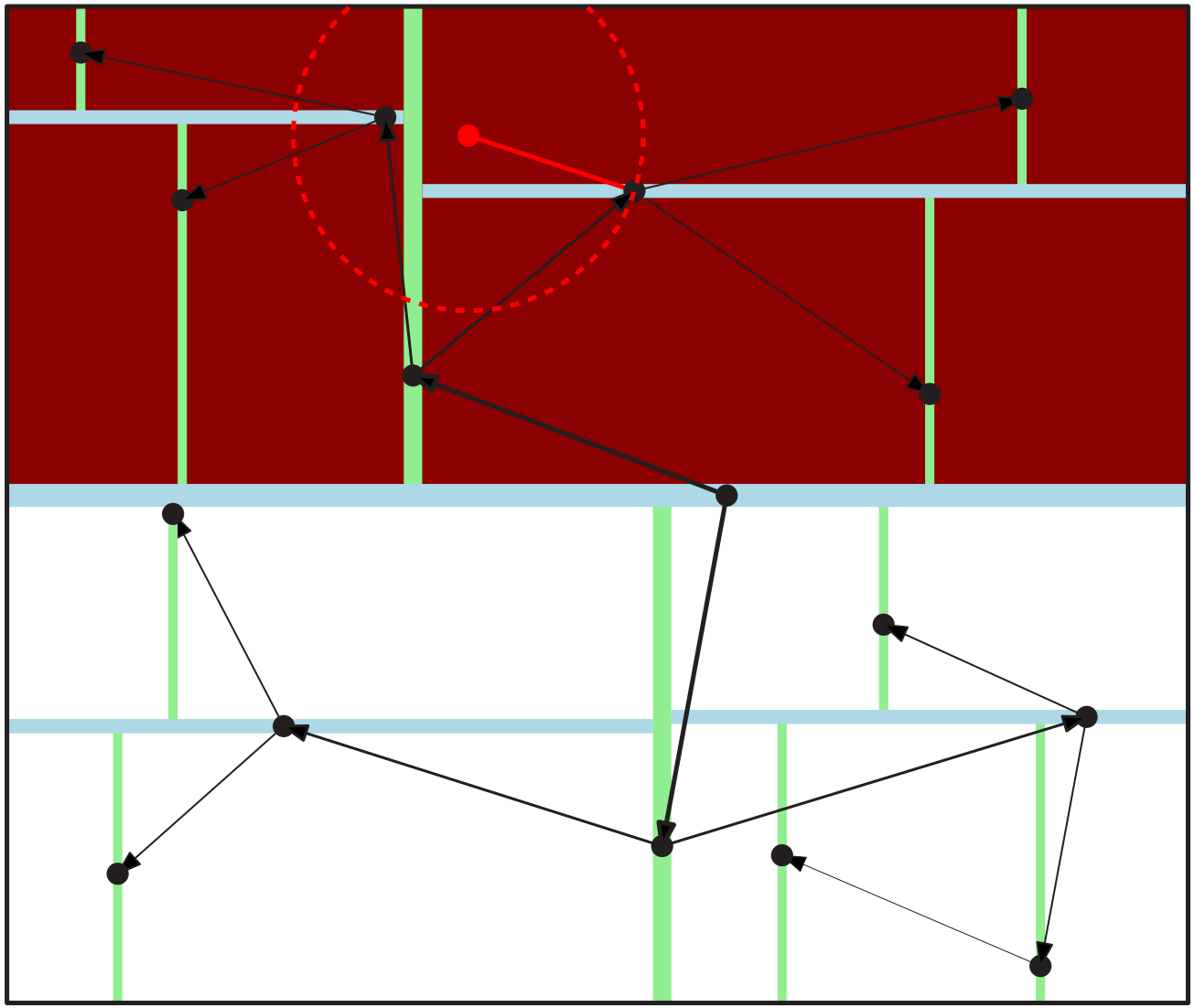
k-d-Tree



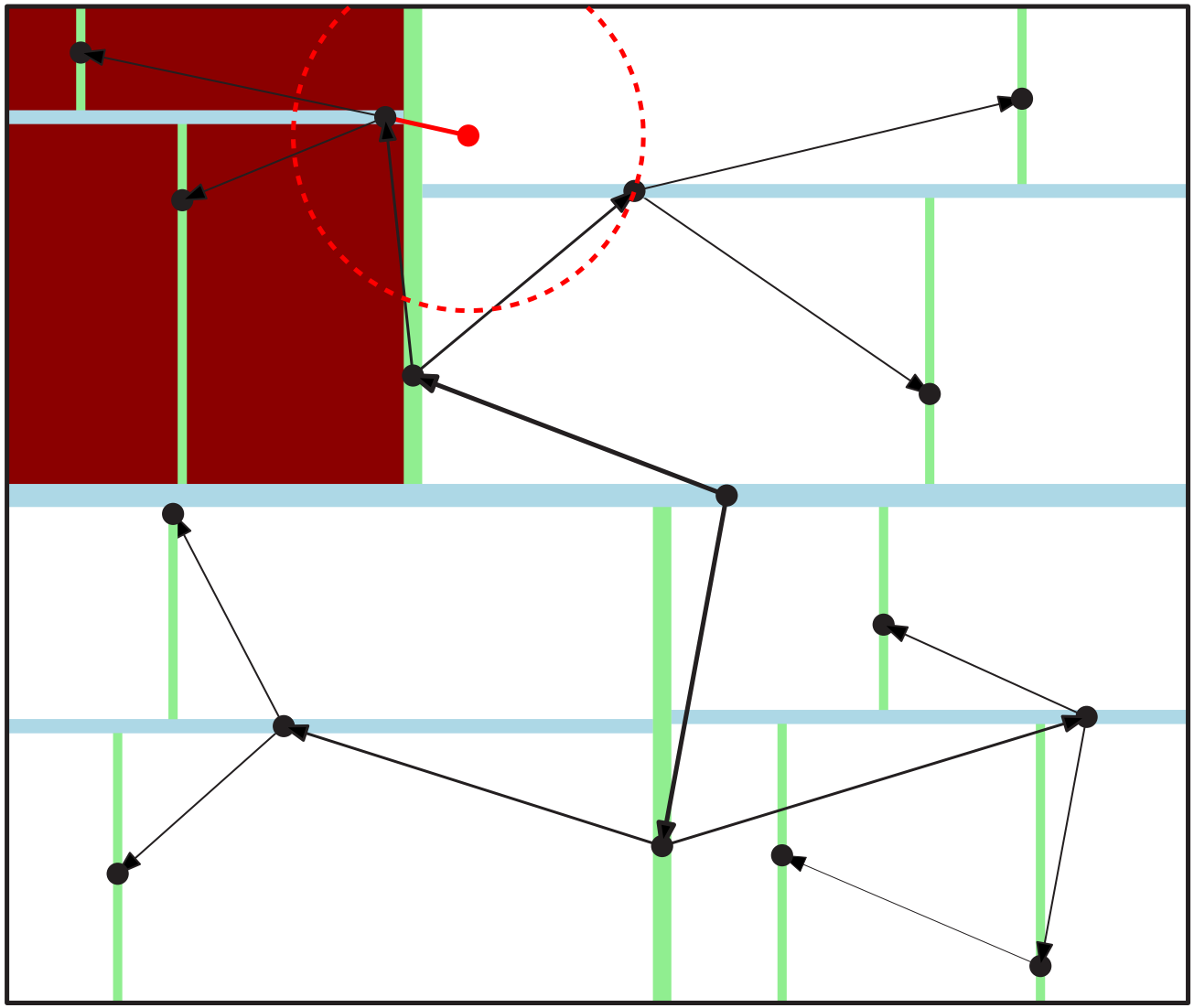
k-d-Tree



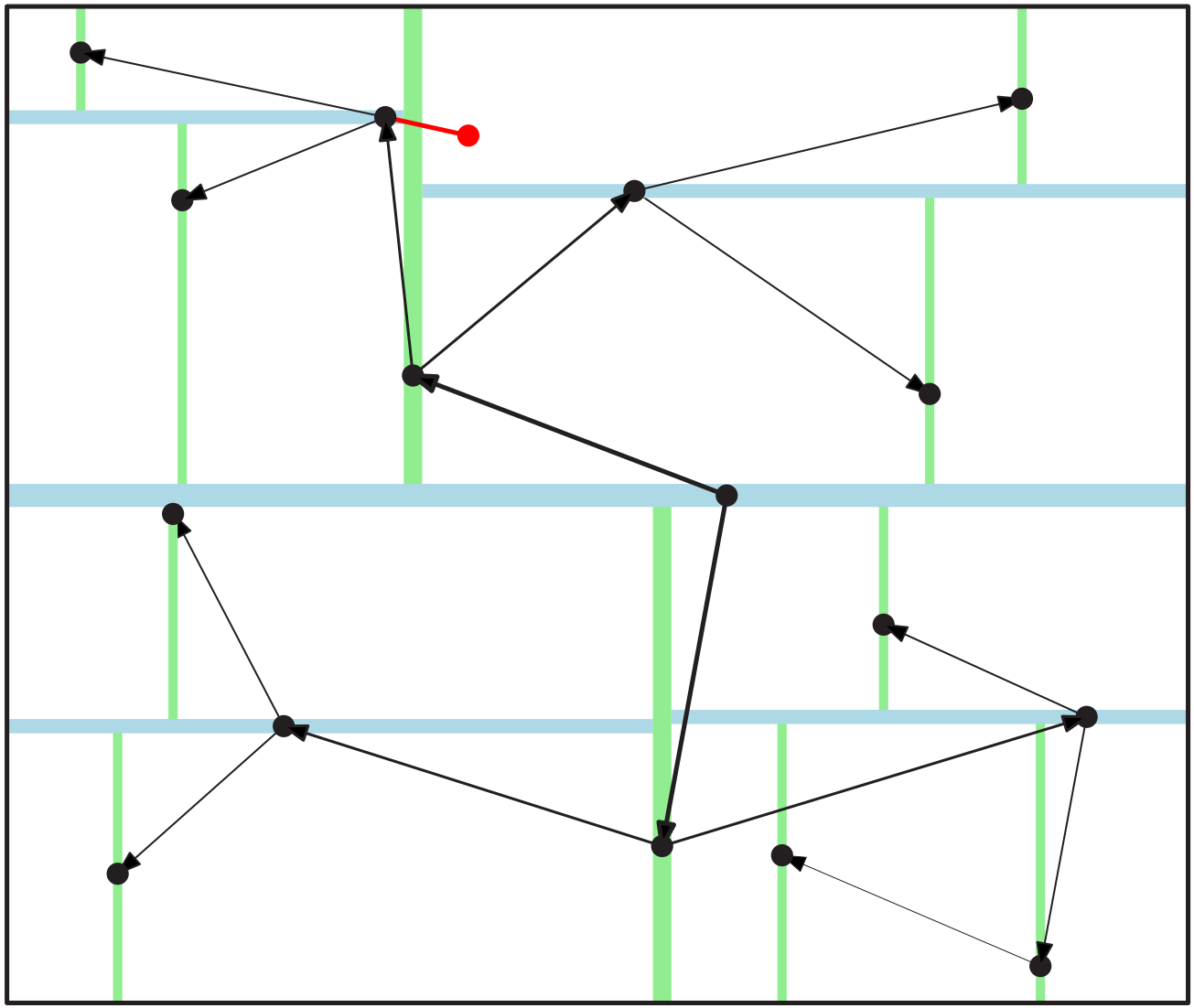
k-d-Tree



k-d-Tree



k-d-Tree



Bitvektoren

Rank und Select

Gegeben ein Bitvektor B der Länge n ($B[i] \in \{0, 1\}$). Wir definieren die folgenden Funktionen auf dem Bitvektor B:

- $rank_1(i, B) =$ Anzahl an 1's in $B[0..i]$
- $select_1(i, B) =$ Position der i-ten 1 in B

$\Rightarrow$ *rank* und *select* können mit $o(n)$ Bits zusätzlichen Speicherplatz und $O(1)$ Laufzeit implementiert werden.

Bitvektoren

Implementierung von Rank

1110 1010 0011 1100 1101 0001 0011 1110 0111 1111 0011

Bitvektoren

Implementierung von Rank

Superblöcke $L = \log^2 n$

1110 1010 0011 1100	1101 0001 0011 1110	0111 1111 0011
0	**9**	**18**

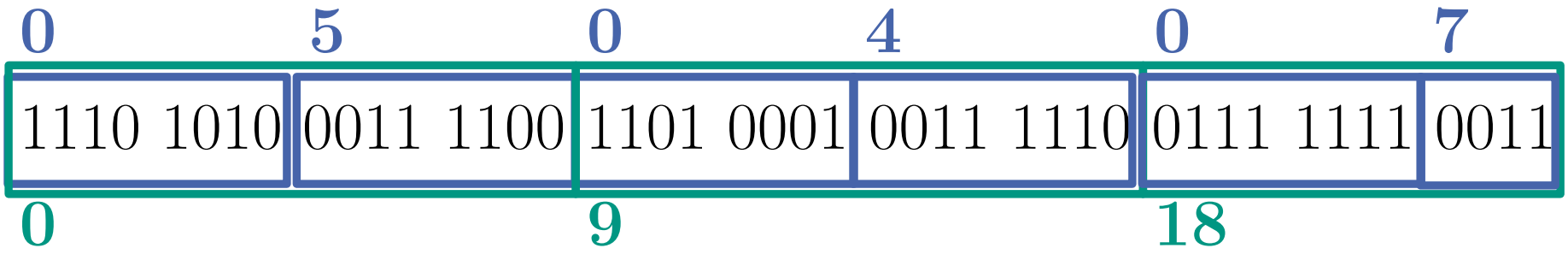
$$Space(rank) = \frac{n}{log^2(n)} \cdot log(n)$$

Bitvektoren

Implementierung von Rank

Superblöcke $L = \log^2 n$

Subblöcke $B = \frac{\log n}{2}$



$$Space(rank) = \frac{n}{log^2(n)} \cdot log(n) + \frac{2n}{\log(n)} \cdot \log\log^2(n)$$

Bitvektoren

Implementierung von Rank

Superblöcke $L = \log^2 n$

Subblöcke $B = \frac{\log n}{2}$

Lookup-Table für *rank* für jede mögliche Position i und beliebiger Bitvektor der Größe $\frac{\log n}{2}$

0	5	0	4	0	7
1110 1010	0011 1100	1101 0001	0011 1110	0111 1111	0011

0 9 18

$$Space(rank) = \frac{n}{log^2(n)} \cdot log(n) + \frac{2n}{\log(n)} \cdot \log\log^2(n) + 2^{\frac{\log n}{2}} \cdot \frac{\log n}{2} \cdot \log\log(n)$$

Bitvektoren

Implementierung von Rank

Superblöcke $L = \log^2 n$

Subblöcke $B = \frac{\log n}{2}$

Lookup-Table für *rank* für jede mögliche Position i und beliebiger Bitvektor der Größe $\frac{\log n}{2}$

0	5	0	4	0	7
1110 1010	0011 1100	1101 0001	0011 1110	0111 1111	0011

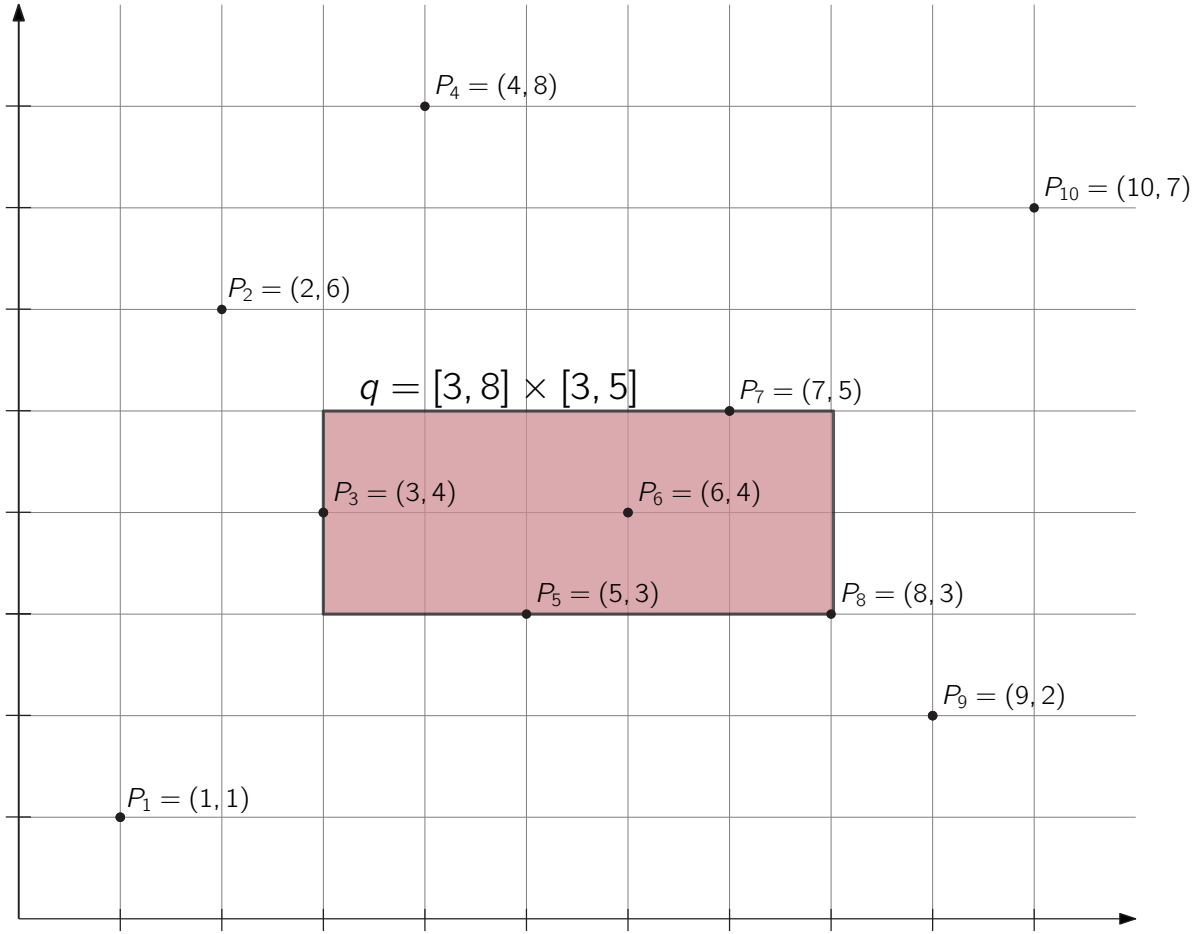
0 9 18

$$Space(rank) = \frac{n}{log^2(n)} \cdot log(n) + \frac{2n}{\log(n)} \cdot \log\log^2(n) + 2^{\frac{\log n}{2}} \cdot \frac{\log n}{2} \cdot \log\log(n)$$

$$= O(\frac{n}{log(n)}) + O(\frac{n}{\log(n)} \cdot \log\log(n)) + O(\sqrt{n} \cdot \log n \cdot \log\log(n)) \in o(n)$$

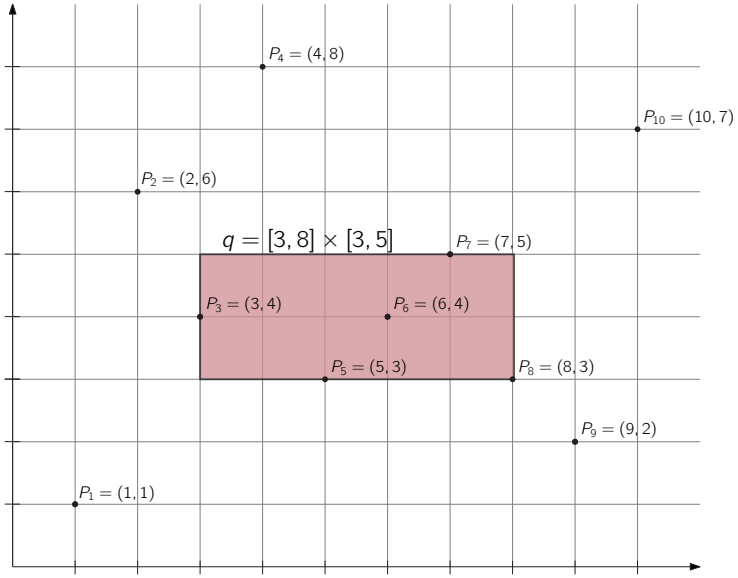
2D Range Queries

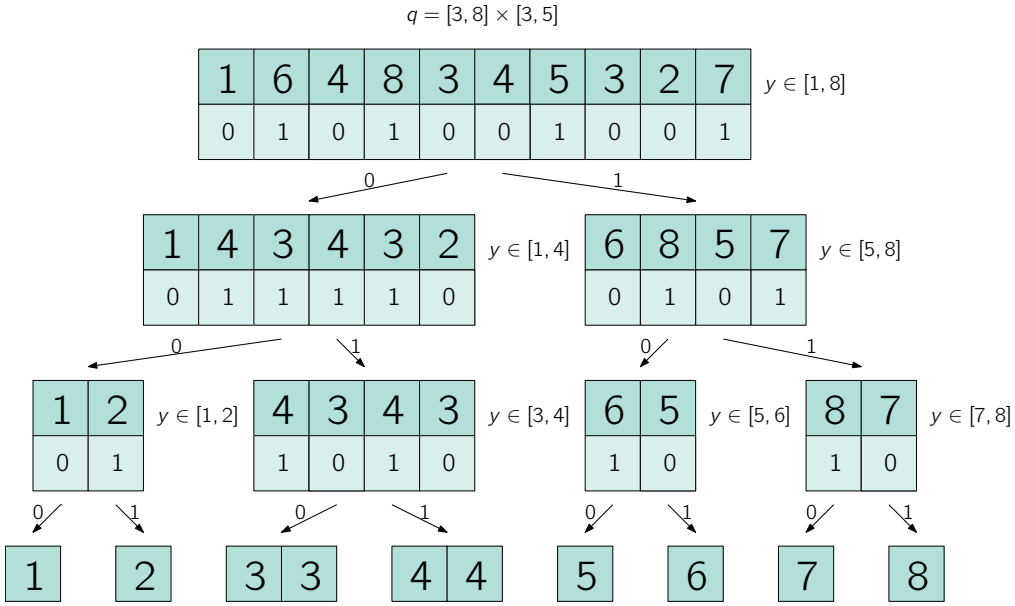
Wavelet Tree



2D Range Queries

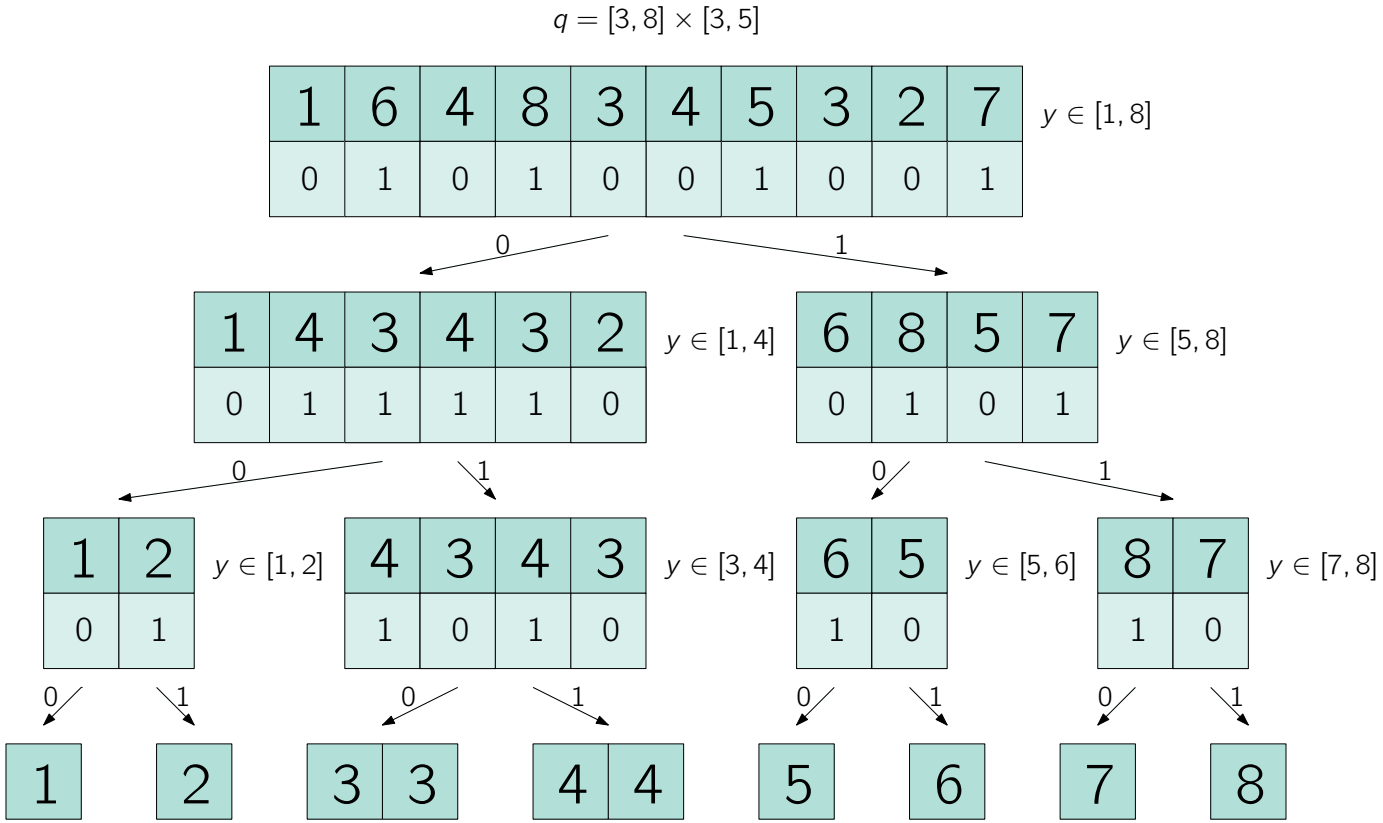
Wavelet Tree





2D Range Queries

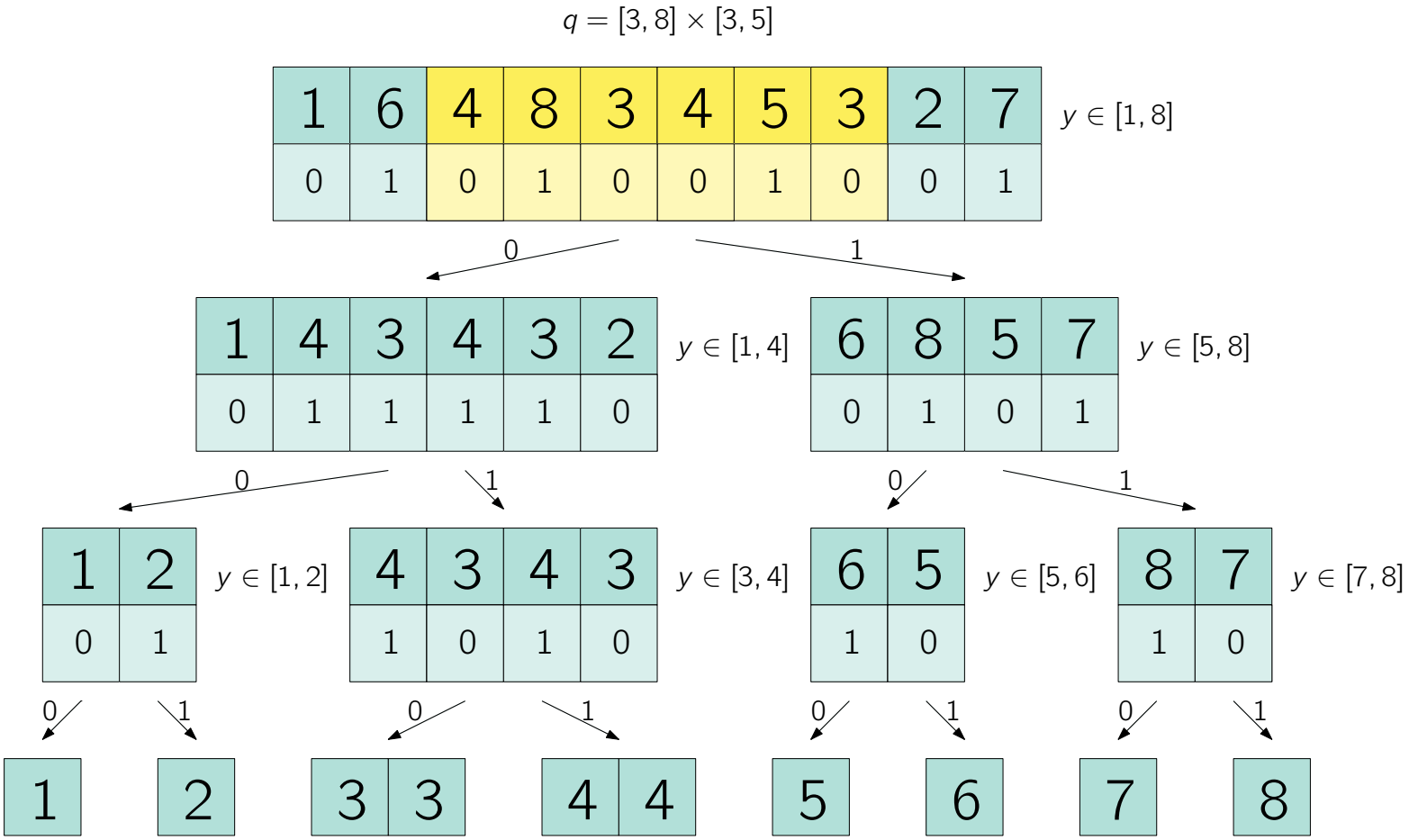
Wavelet Tree



Auf einem $n \times n$ Grid braucht ein Wavelet Tree $n \log n + o(n \log n)$ Bits Speicherplatz

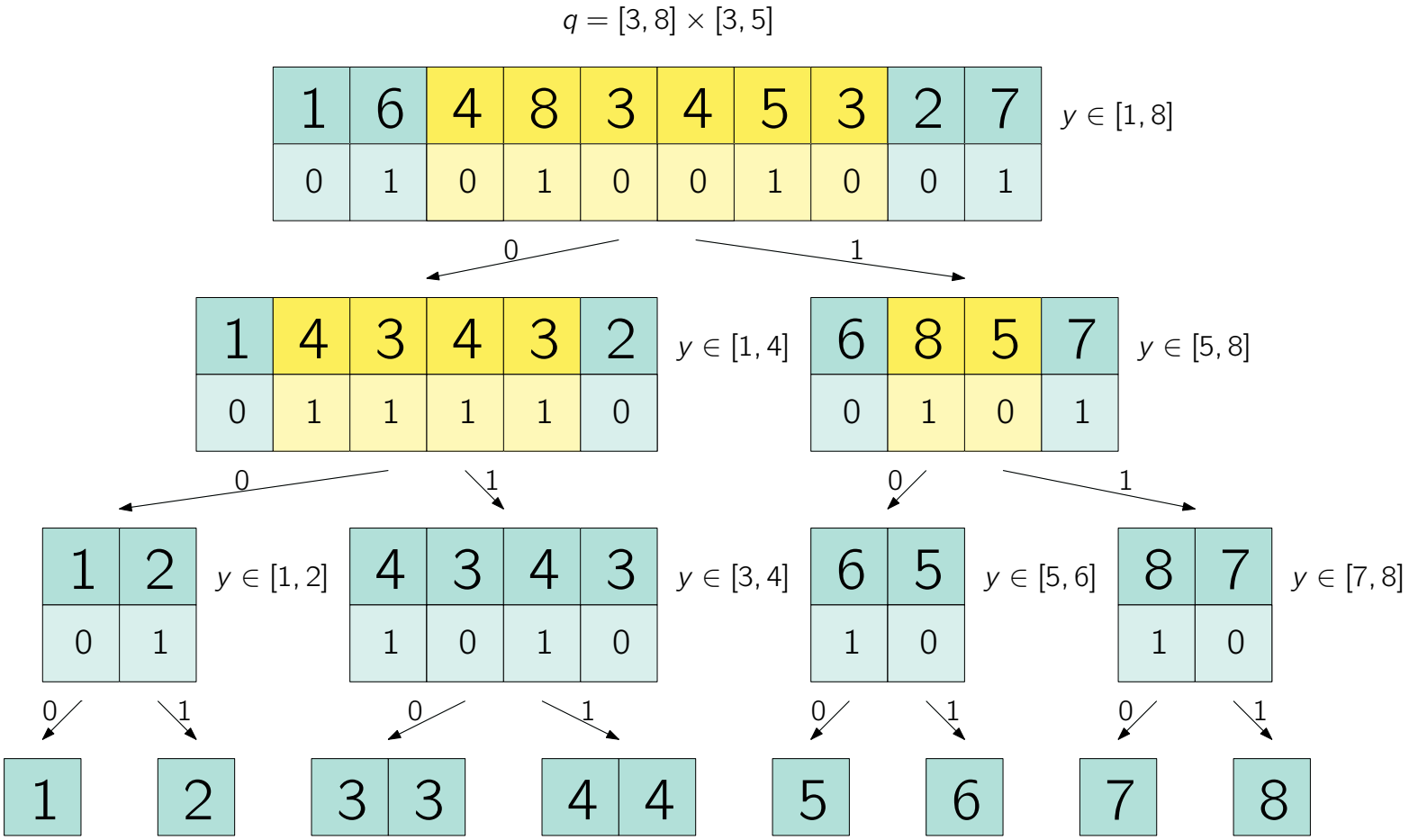
2D Range Queries

Wavelet Tree - Count Operation



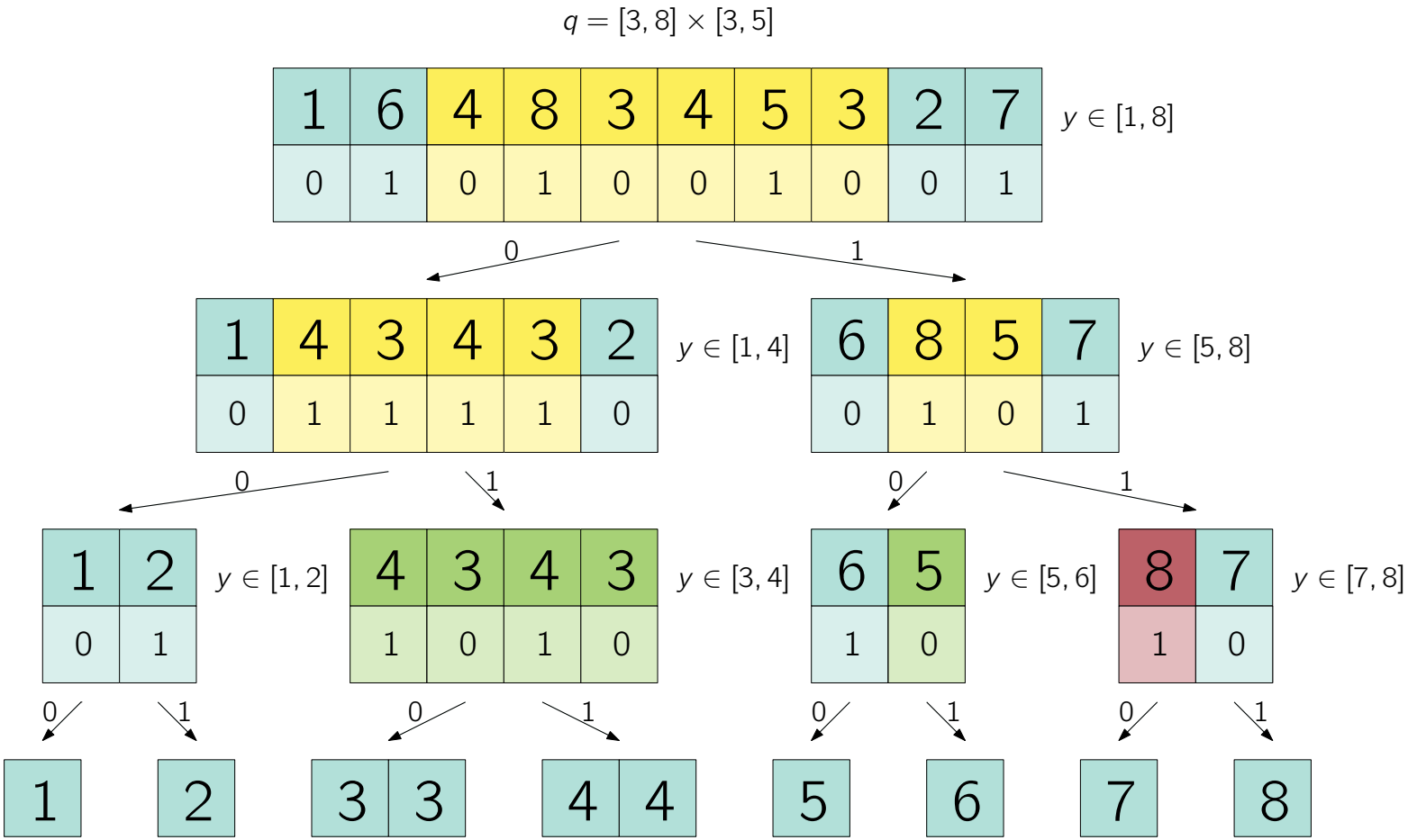
2D Range Queries

Wavelet Tree - Count Operation



2D Range Queries

Wavelet Tree - Count Operation



$\mathcal{O}(\log n)$ Baumknoten werden besucht bevor die Recursion stoppt $\Rightarrow$ count hat Laufzeit $\mathcal{O}(\log n)$

2D Range Queries

K^2-Treaps

5			8	5		7	6
1		2		2	3	4	1
	7	4	2				
7	3		1				
				7			
						3	2
						1	0

2D Range Queries

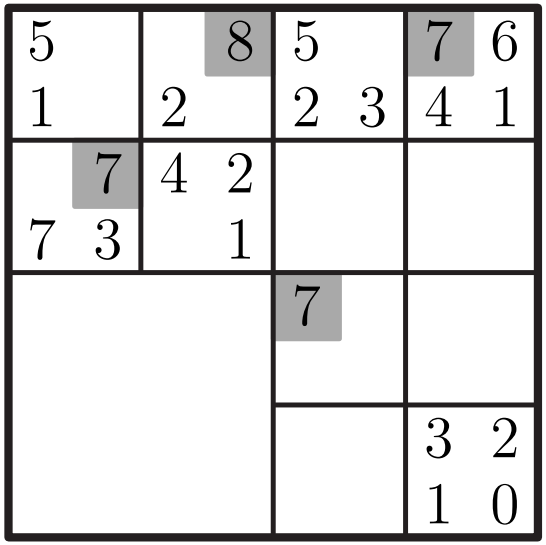
K^2-Treaps

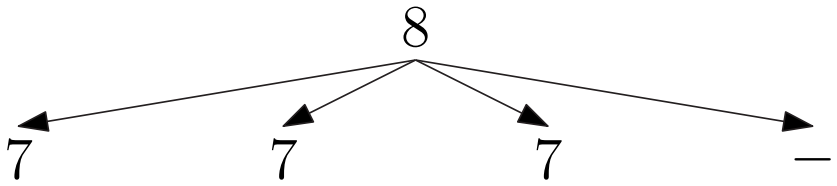
5			8	5		7	6
1		2		2	3	4	1
	7	4	2				
7	3		1				
				7			
						3	2
						1	0

8

2D Range Queries

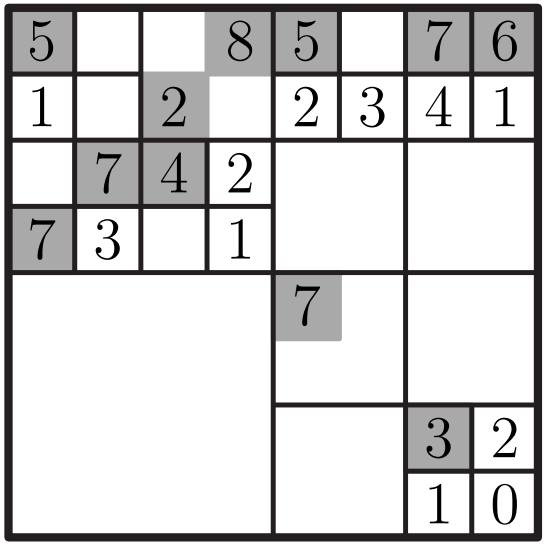
K^2-Treaps

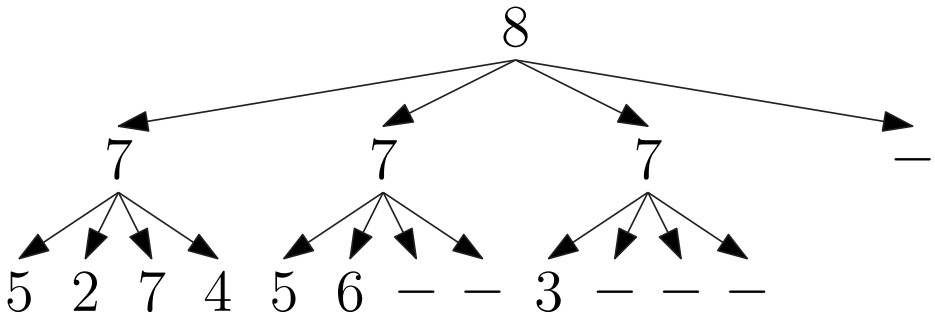




2D Range Queries

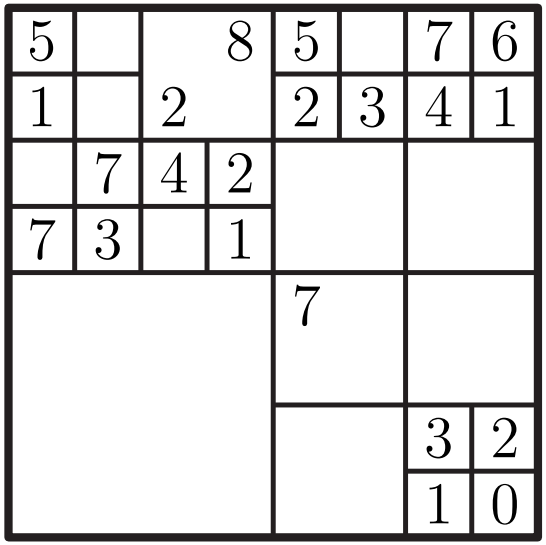
K^2-Treaps

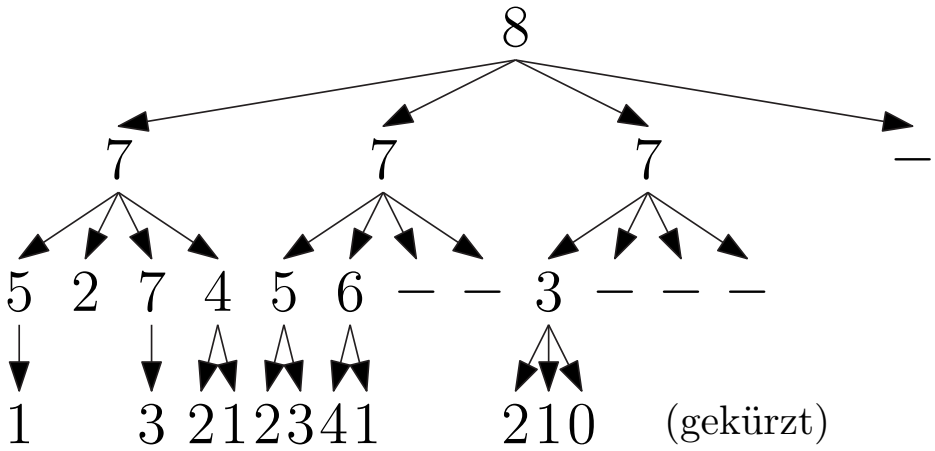




2D Range Queries

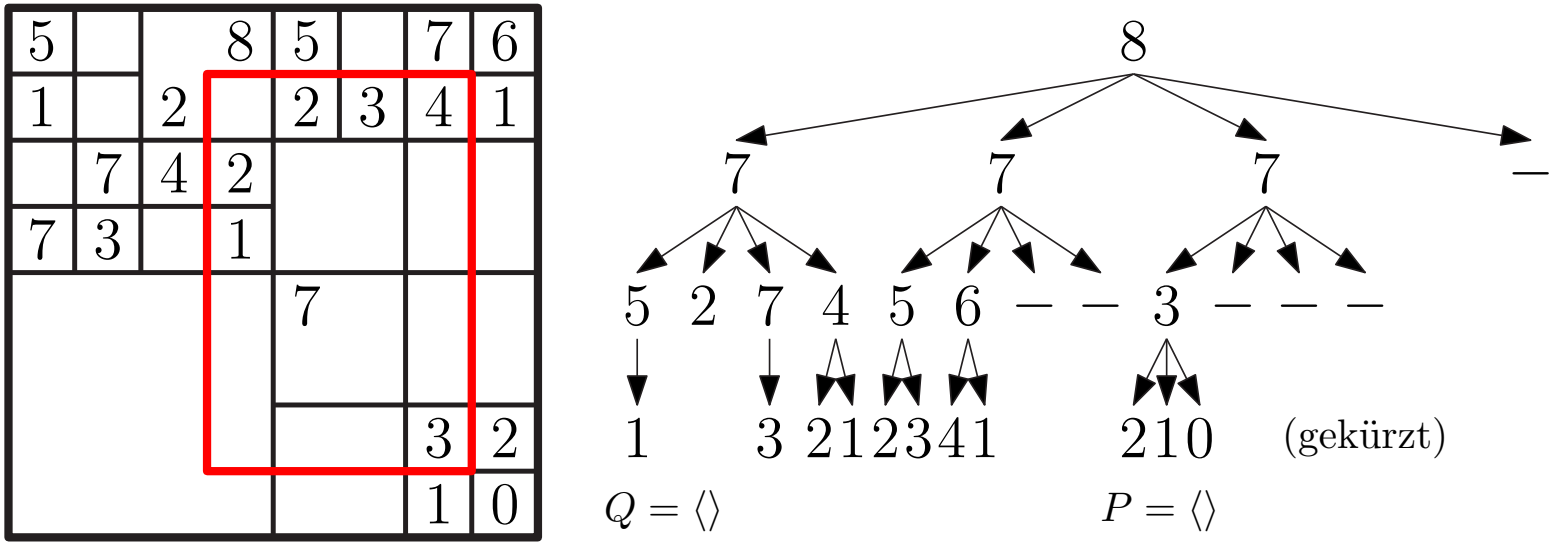
K^2-Treaps





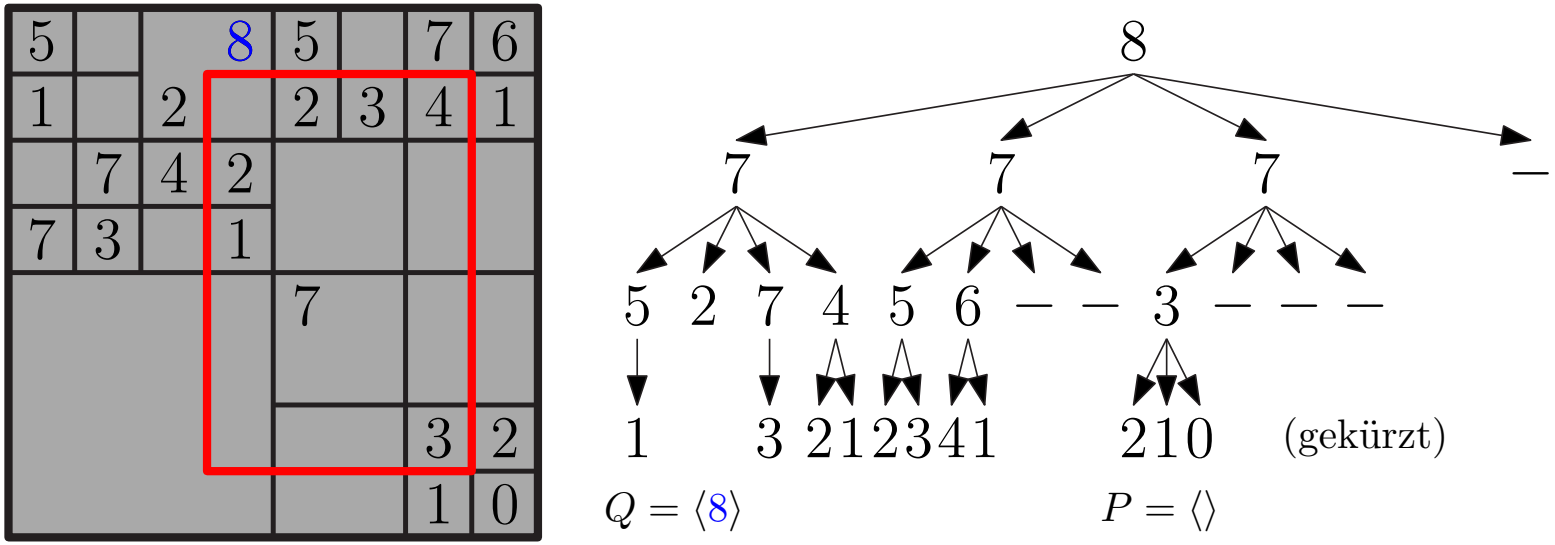
2D Range Queries

K^2-Treaps



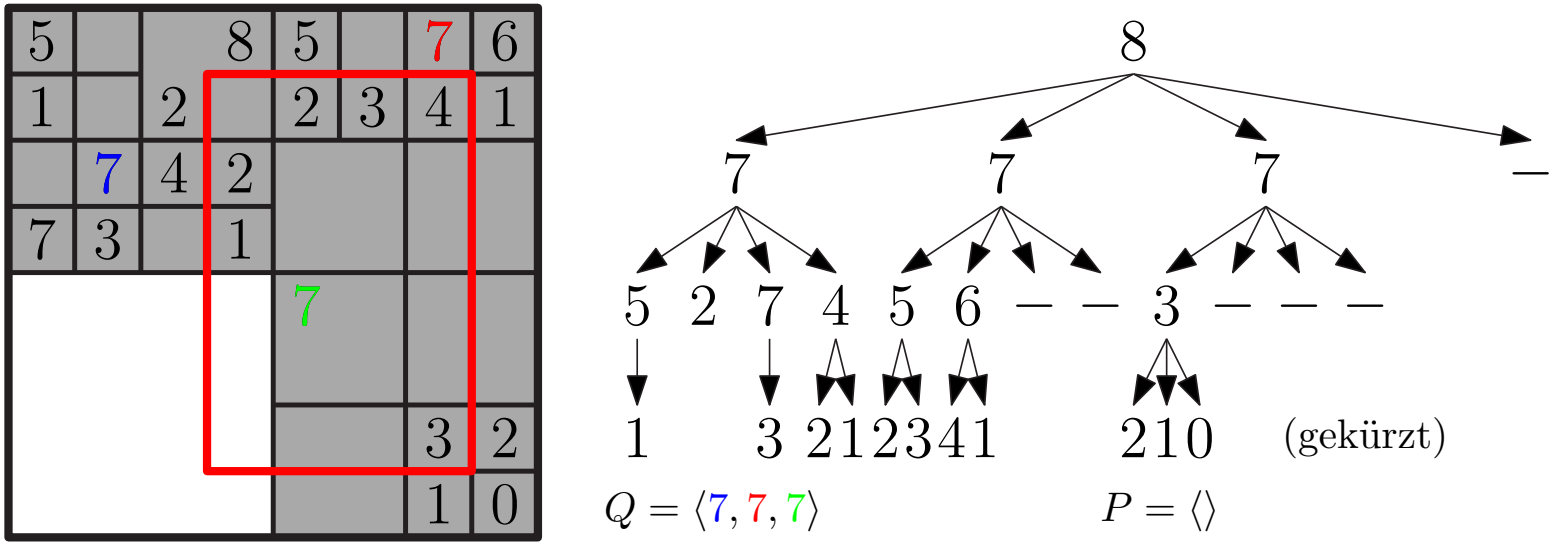
2D Range Queries

K^2-Treaps



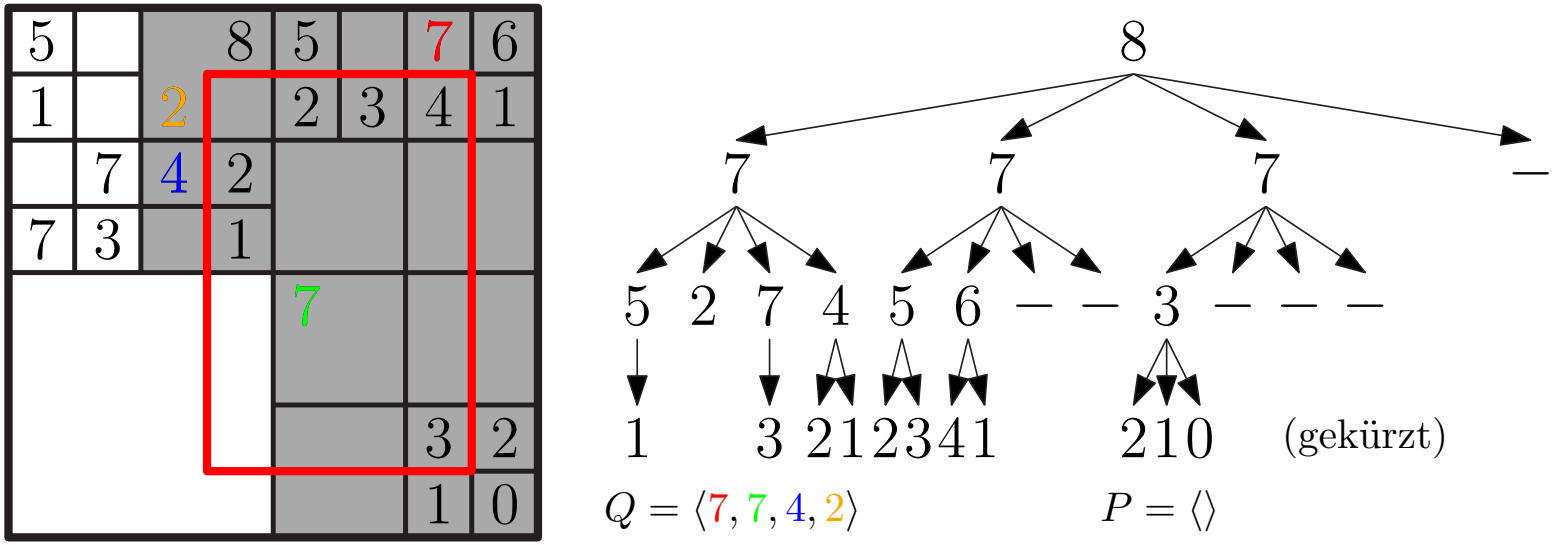
2D Range Queries

K^2-Treaps



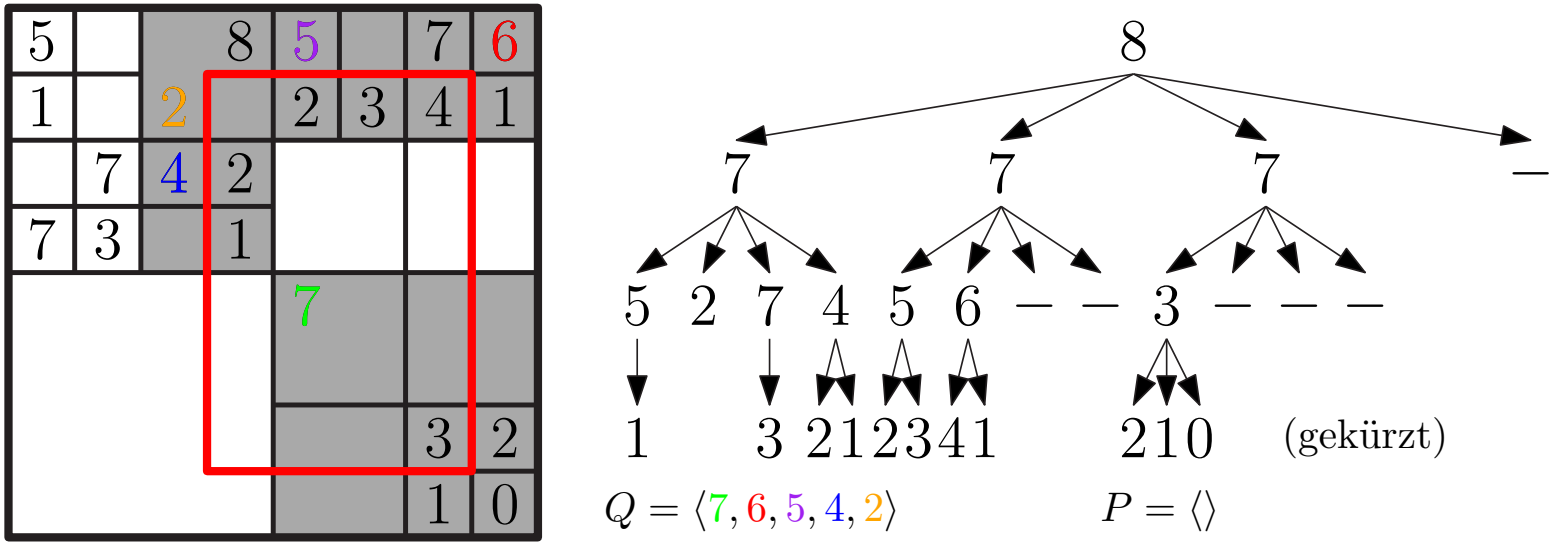
2D Range Queries

K^2-Treaps



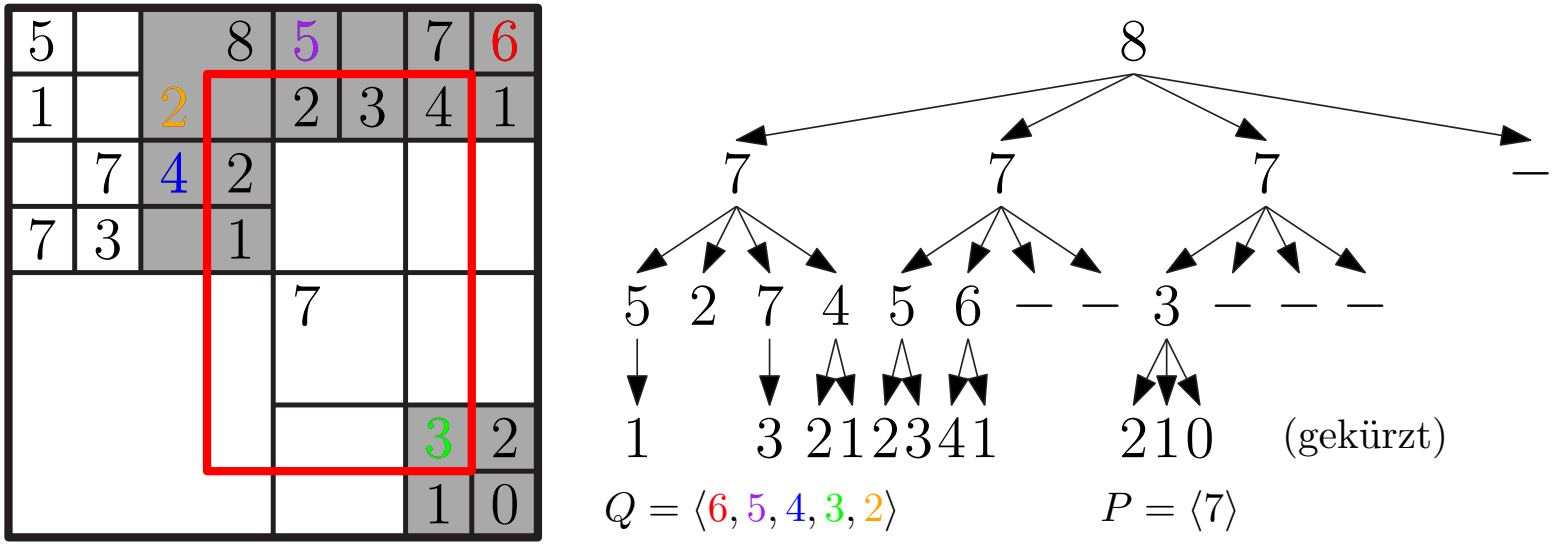
2D Range Queries

K^2-Treaps



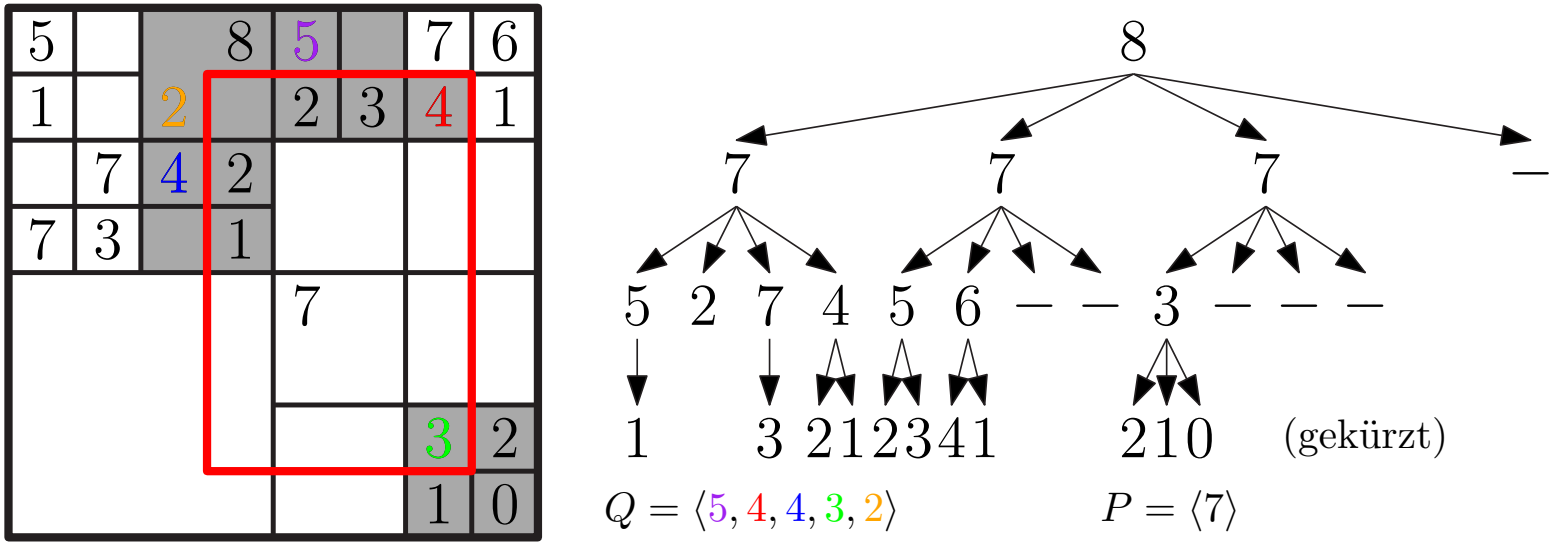
2D Range Queries

K^2-Treaps



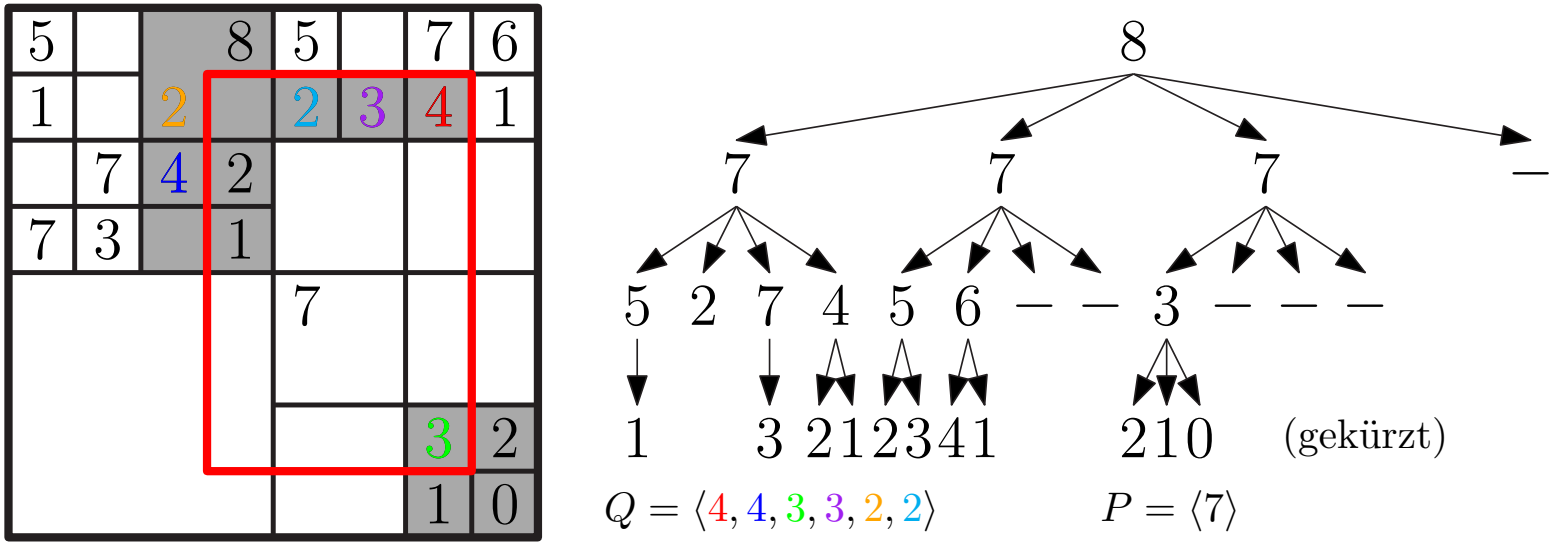
2D Range Queries

K^2-Treaps



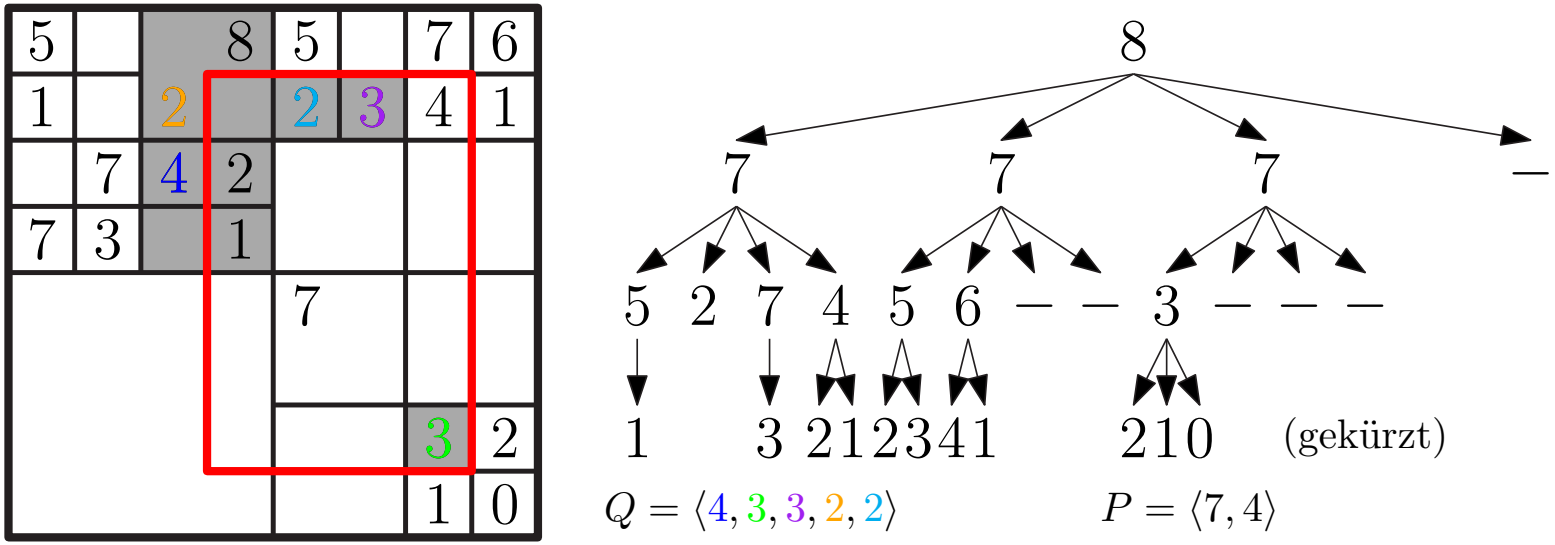
2D Range Queries

K^2-Treaps



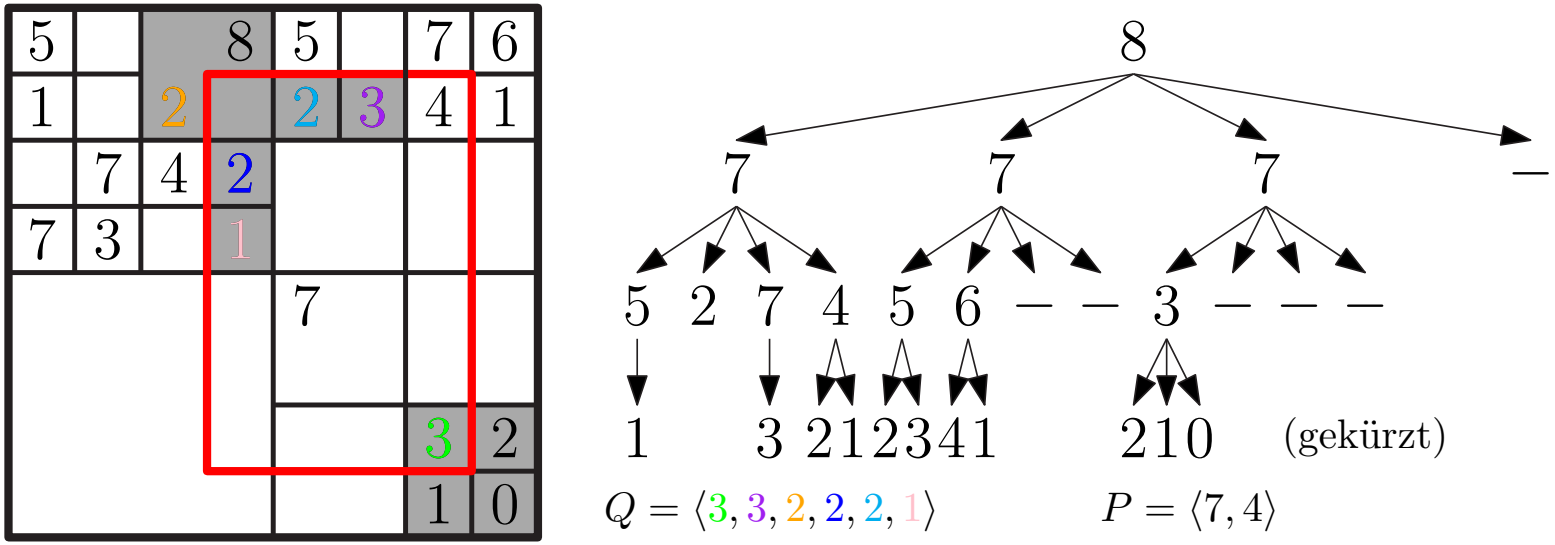
2D Range Queries

K^2-Treaps



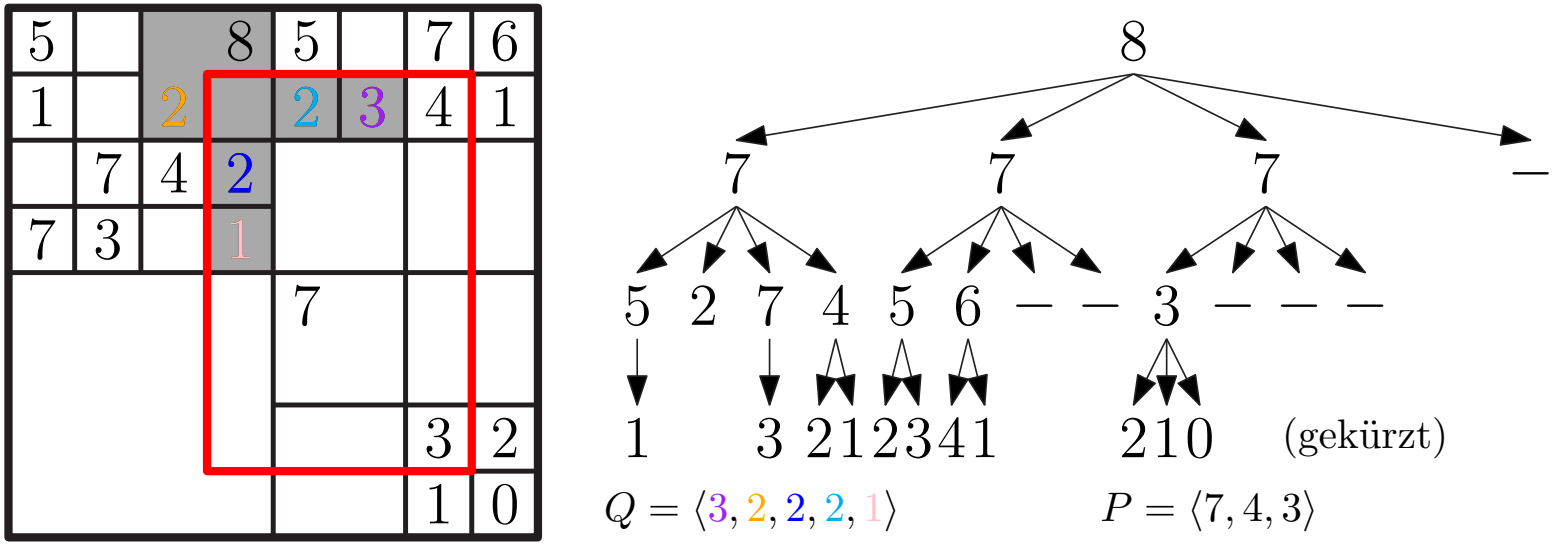
2D Range Queries

K^2-Treaps



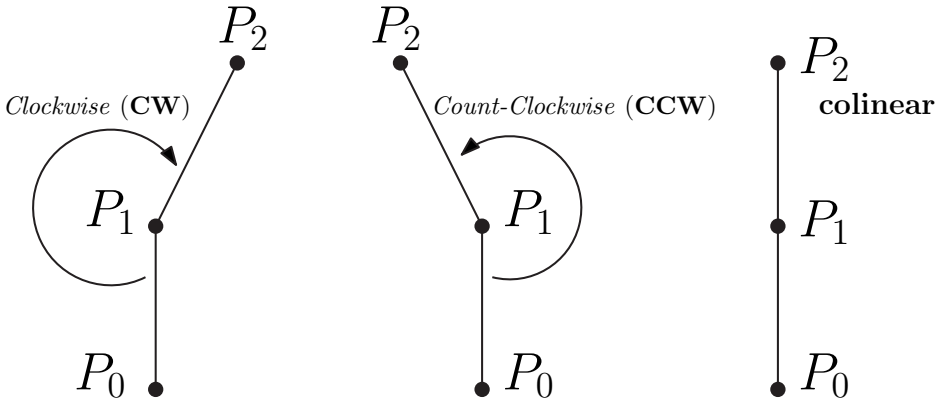
2D Range Queries

K^2-Treaps

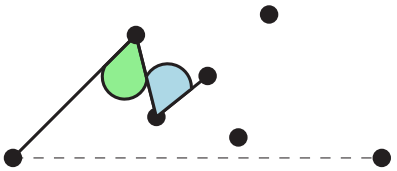


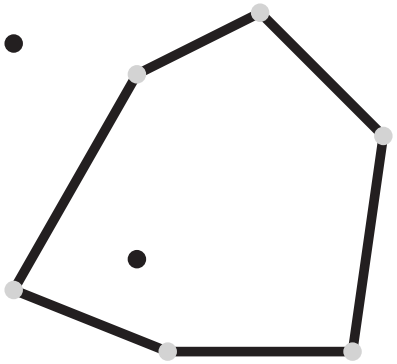
Punktorientierung

- Gegeben drei Punkte P_0, P_1, P_2 $\Rightarrow$ bestimme Orientierung
- Sei $\overrightarrow{a} = \overrightarrow{P_0P_1}$ und $\overrightarrow{b} = \overrightarrow{P_1P_2}$
- $CCW(P_0, P_1, P_2) = a_y \cdot b_x - a_x \cdot b_y$
- $CCW(P_0, P_1, P_2) < 0 \Rightarrow$ **CCW**
- $CCW(P_0, P_1, P_2) = 0 \Rightarrow$ **Colinear**
- $CCW(P_0, P_1, P_2) > 0 \Rightarrow$ **CW**



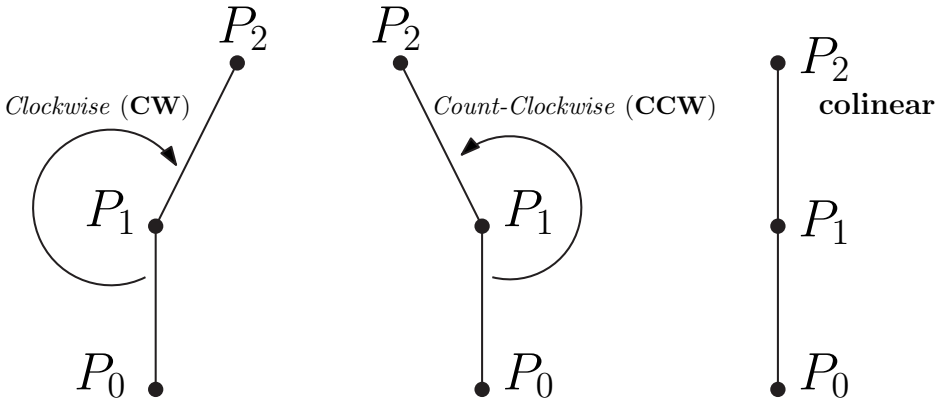
- Unterproblem von Algorithmen
  - Graham Scan
  - Test auf Enthaltensein
    (Punkt in konvexem Polygon)



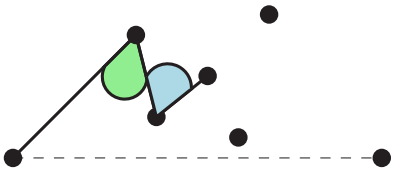


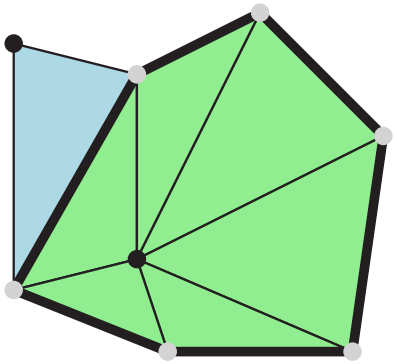
Punktorientierung

- Gegeben drei Punkte P_0, P_1, $P_2 \Rightarrow$ bestimme Orientierung
- Sei $\overrightarrow{a} = \overrightarrow{P_0P_1}$ und $\overrightarrow{b} = \overrightarrow{P_1P_2}$
- $CCW(P_0, P_1, P_2) = a_y \cdot b_x - a_x \cdot b_y$
- $CCW(P_0, P_1, P_2) < 0 \Rightarrow$ **CCW**
- $CCW(P_0, P_1, P_2) = 0 \Rightarrow$ **Colinear**
- $CCW(P_0, P_1, P_2) > 0 \Rightarrow$ **CW**



- Unterproblem von Algorithmen
  - Graham Scan
  - Test auf Enthaltensein
    (Punkt in konvexem Polygon)

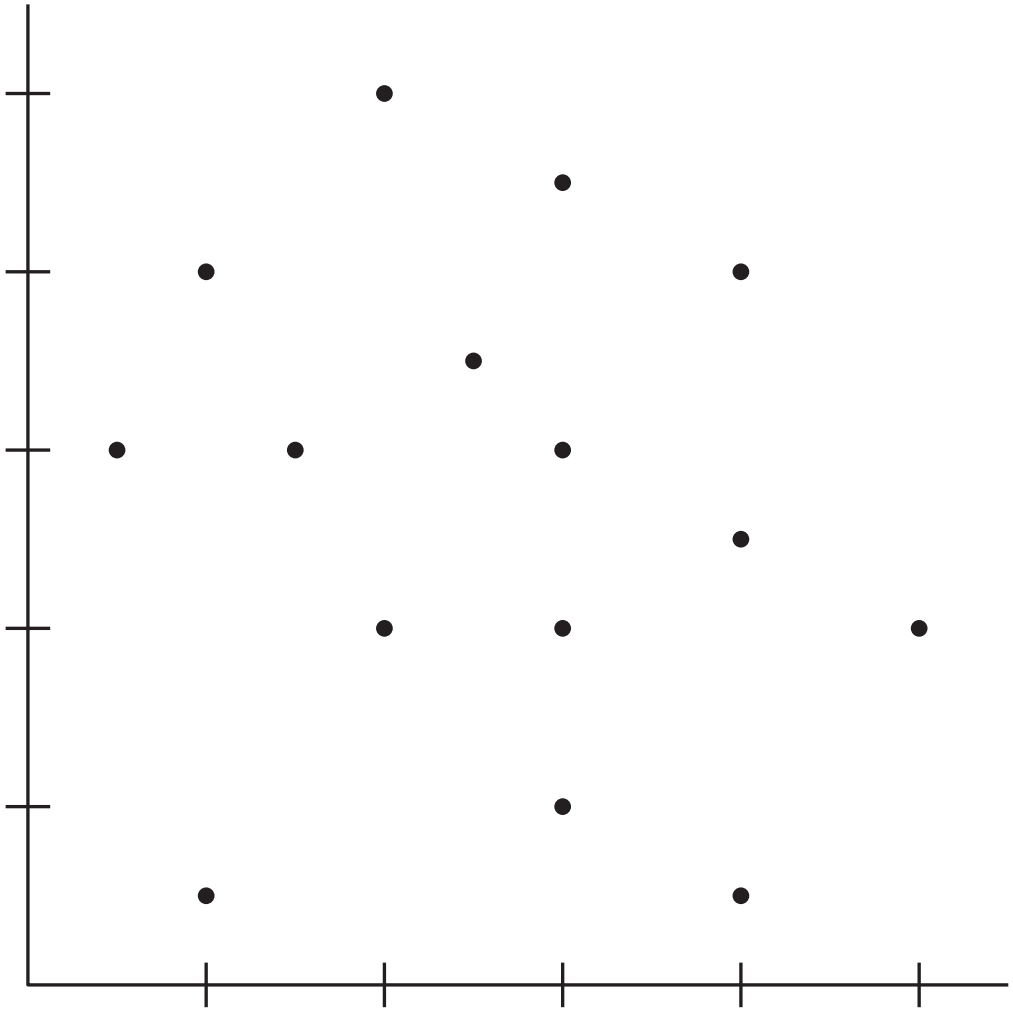




Konvexe Hülle

Graham-Scan

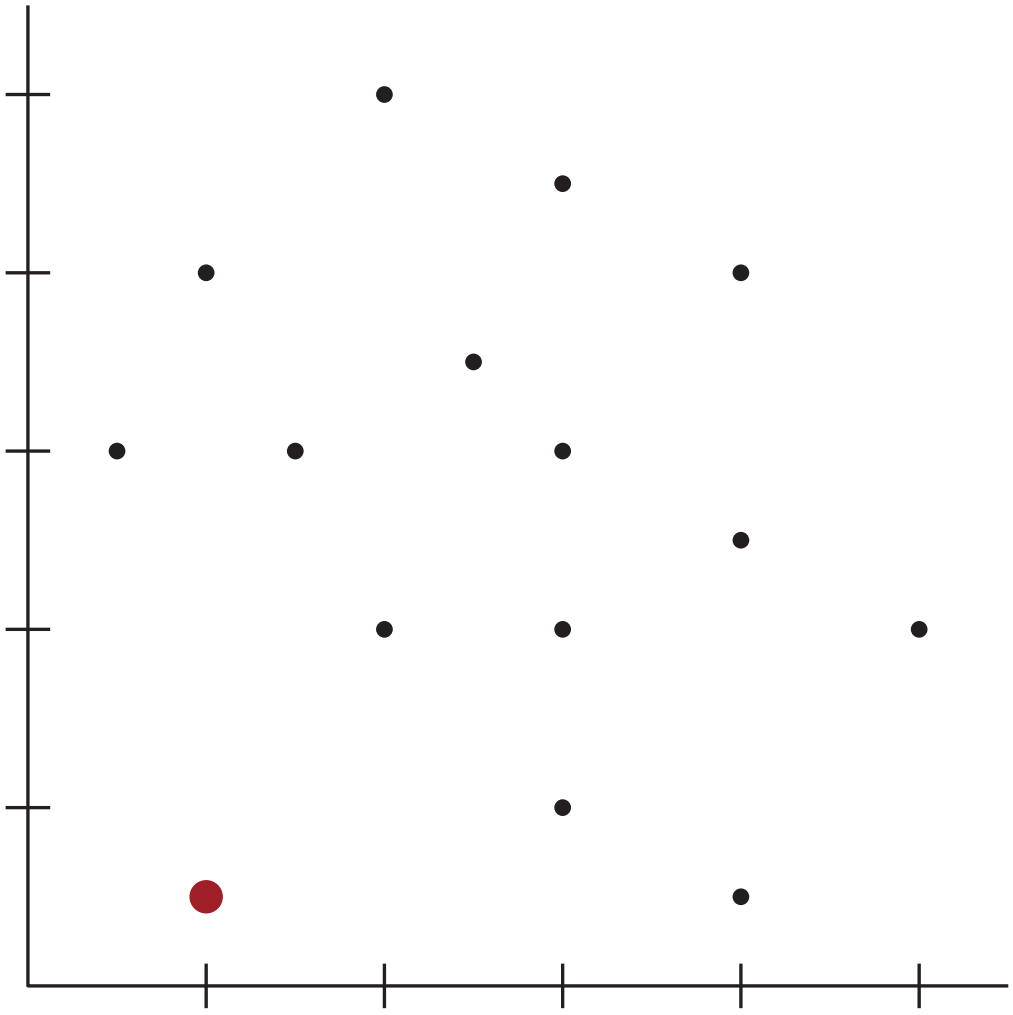
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

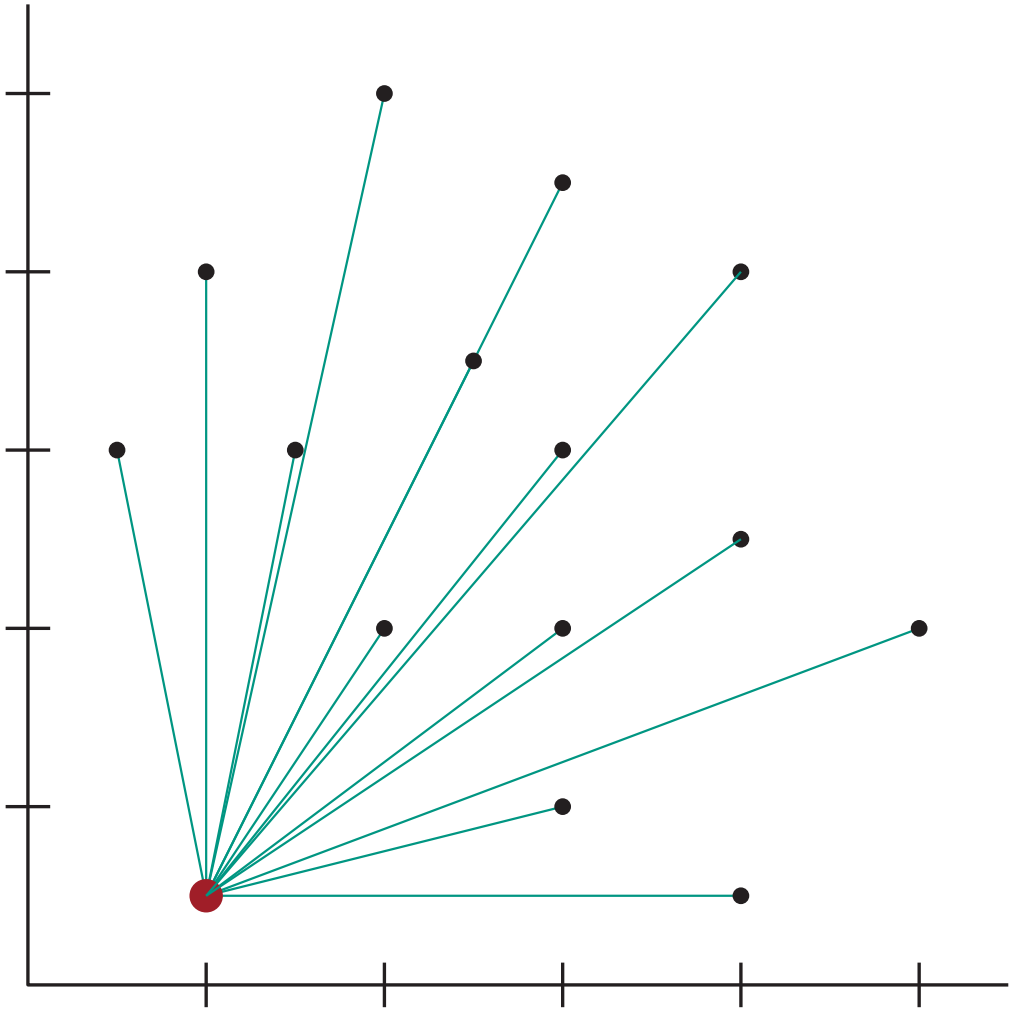
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

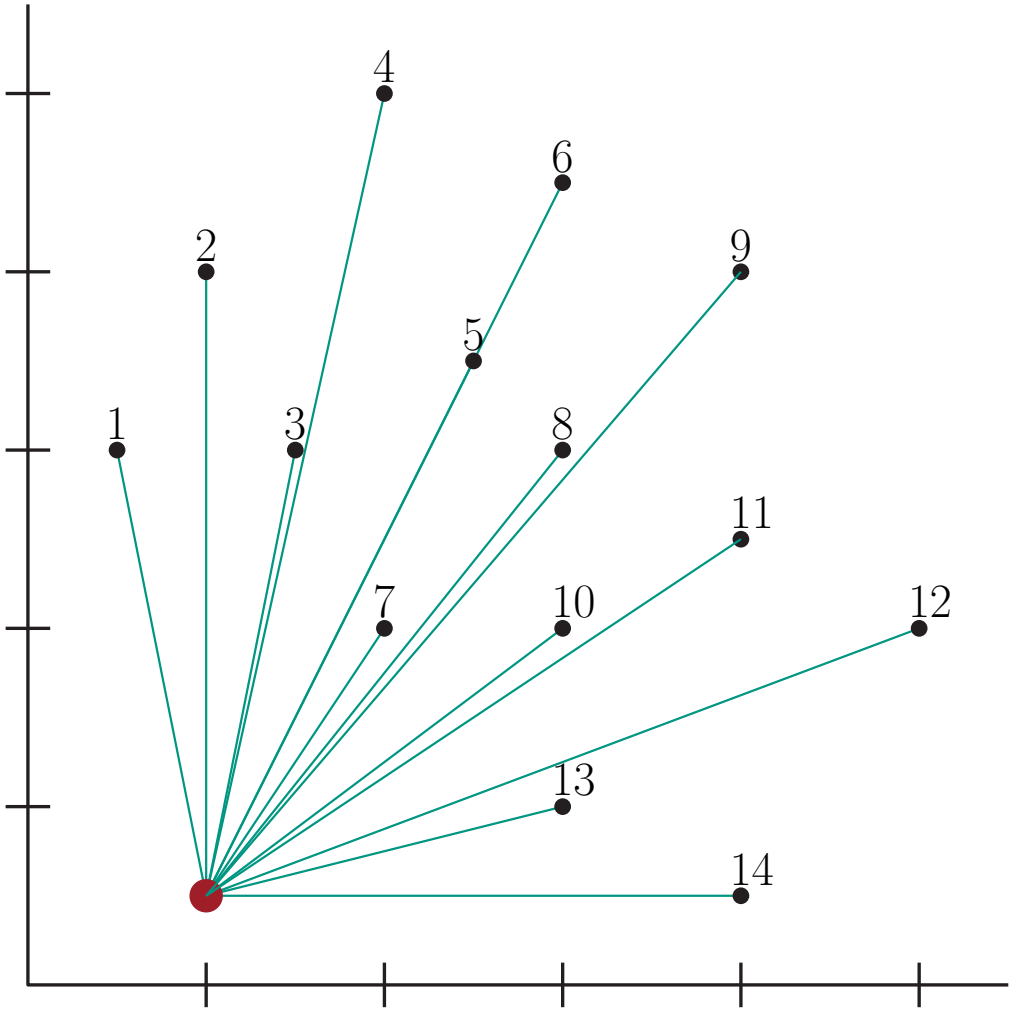
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

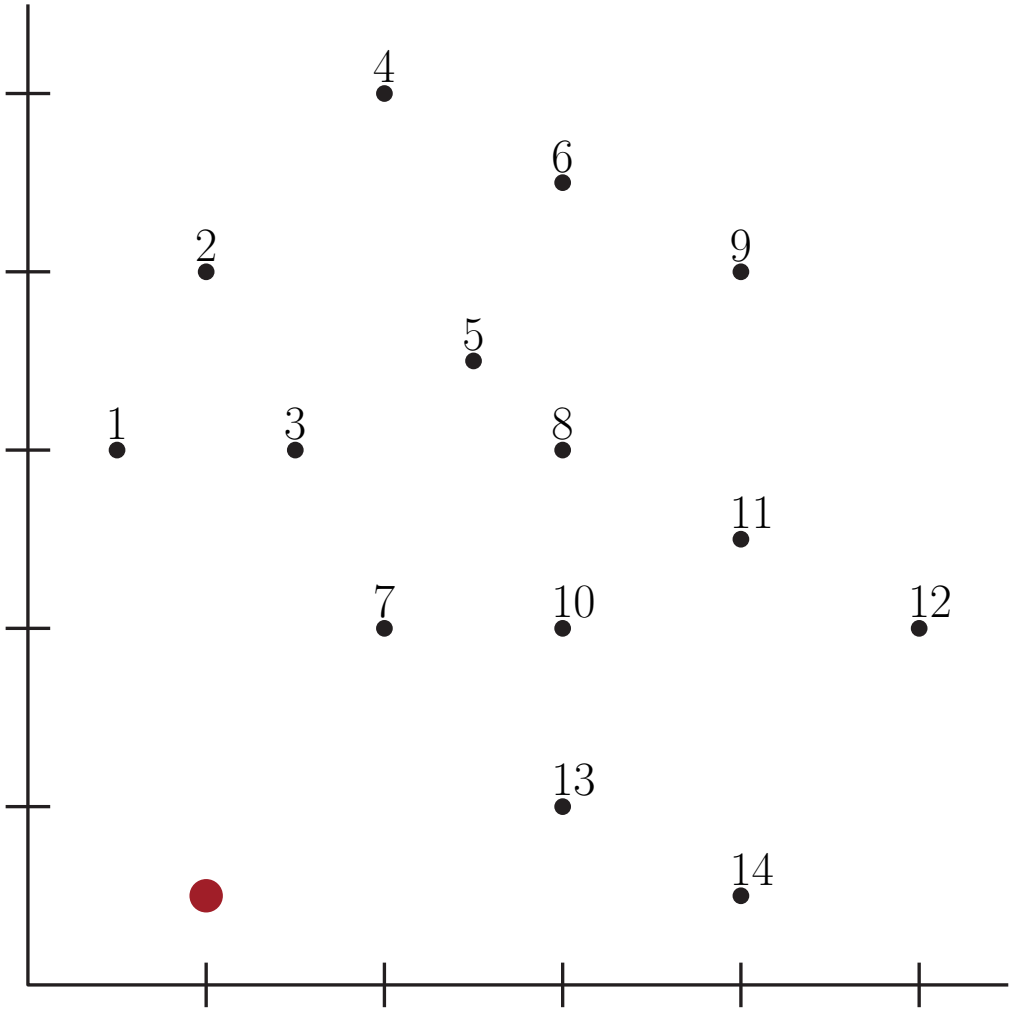
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

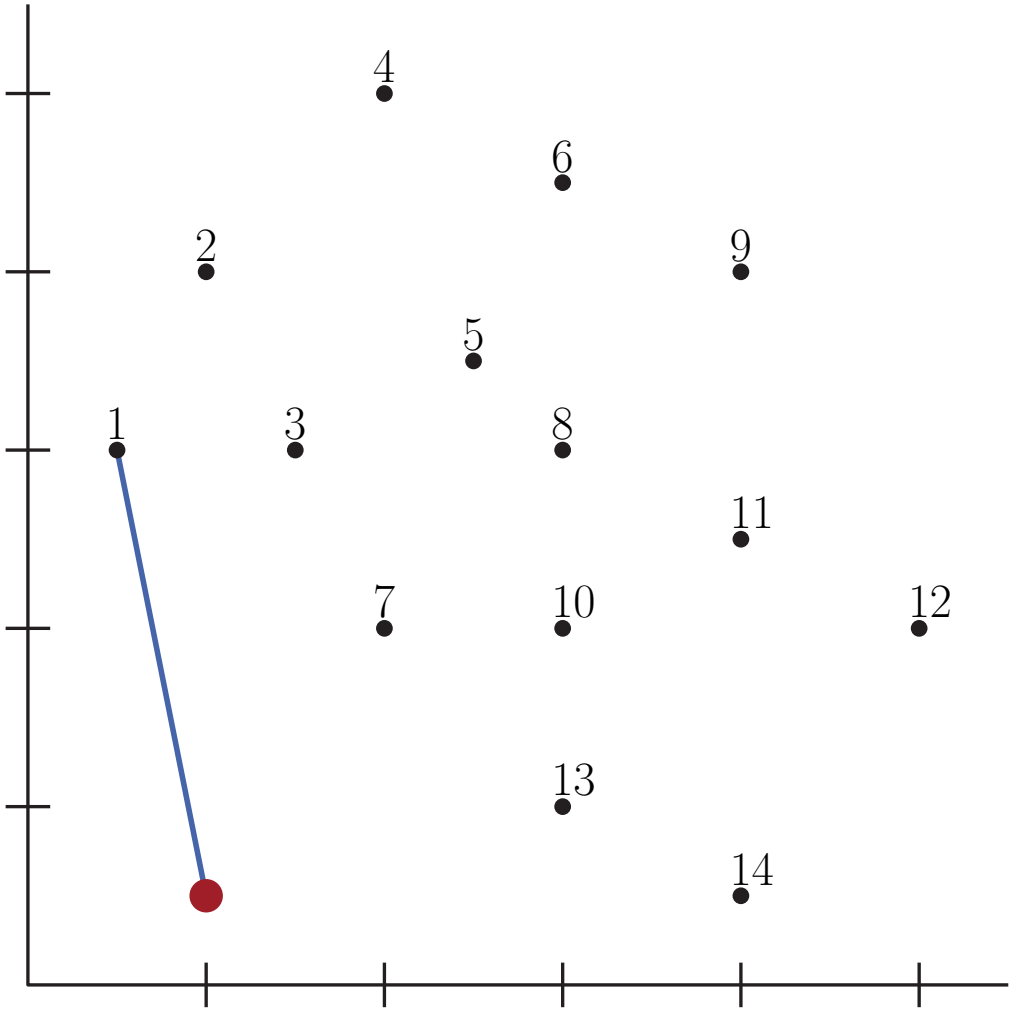
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

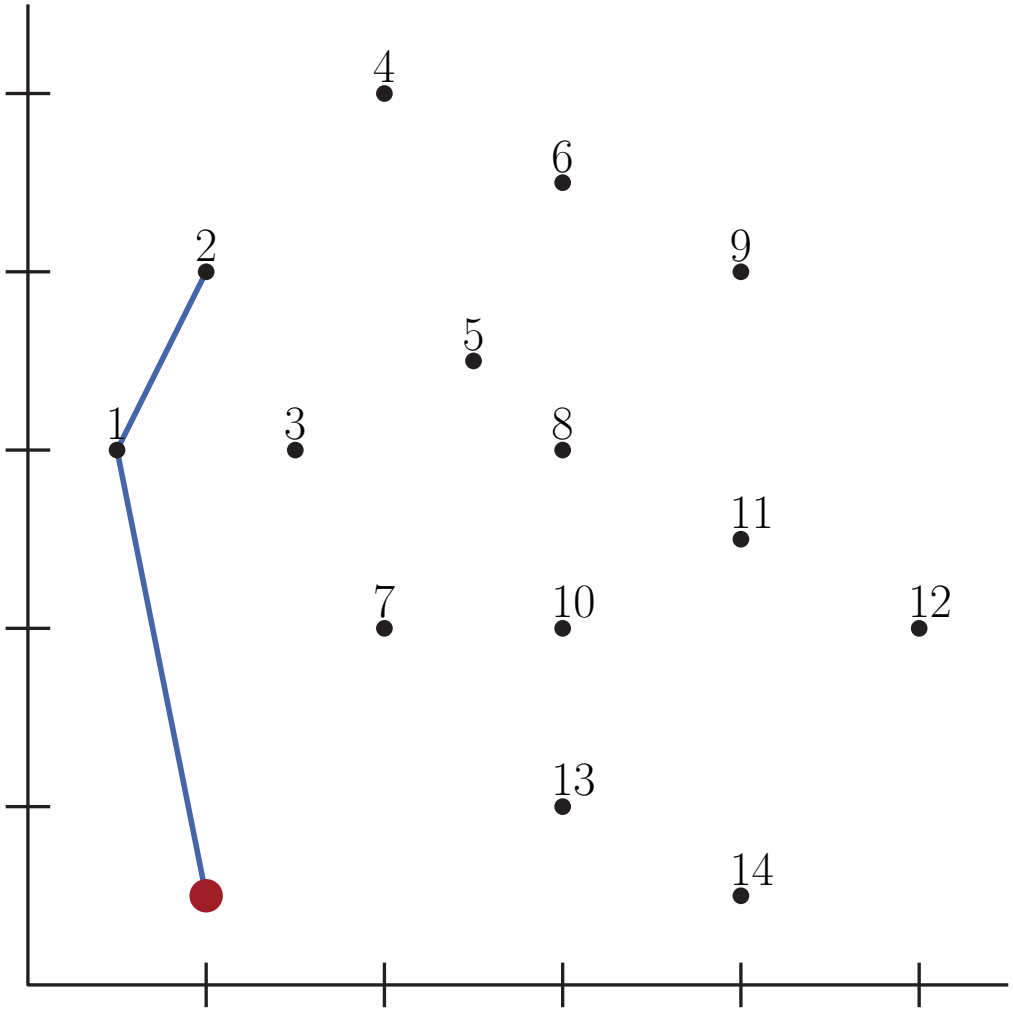
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

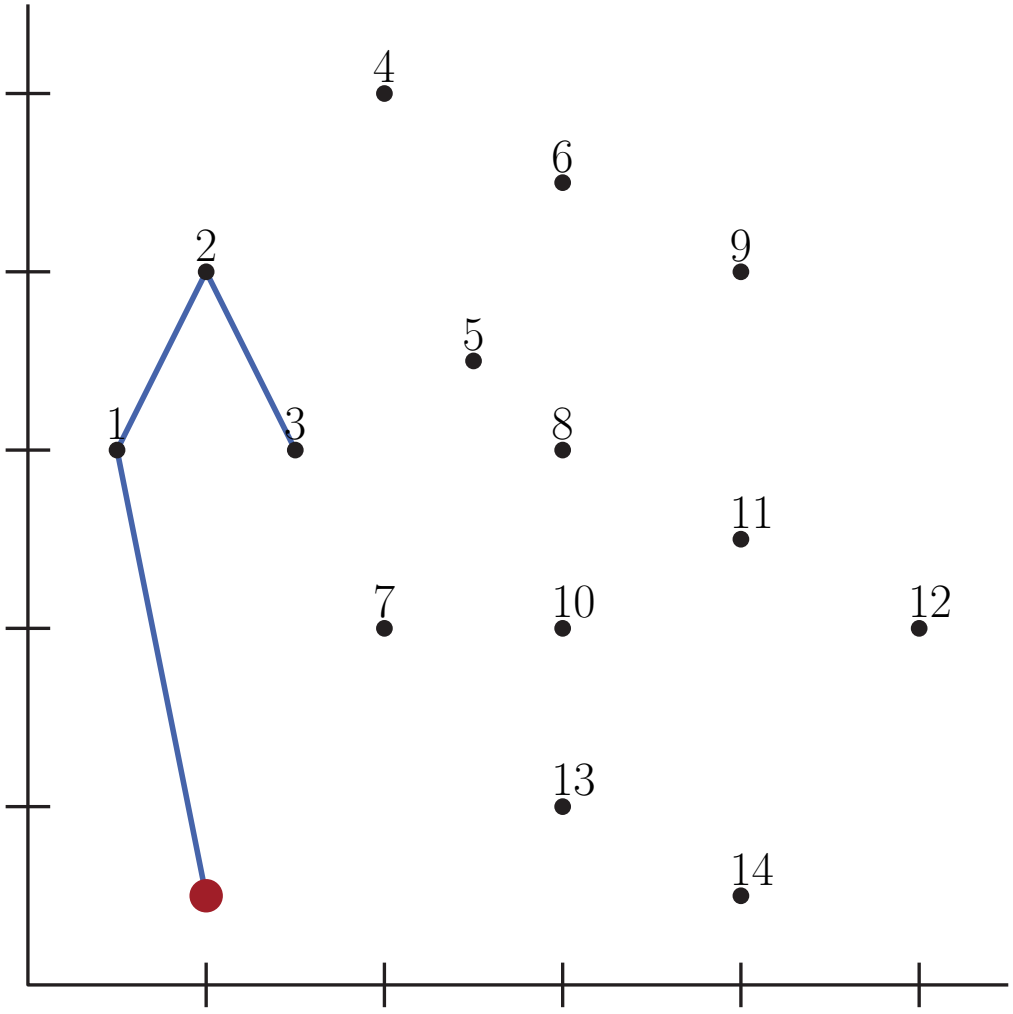
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

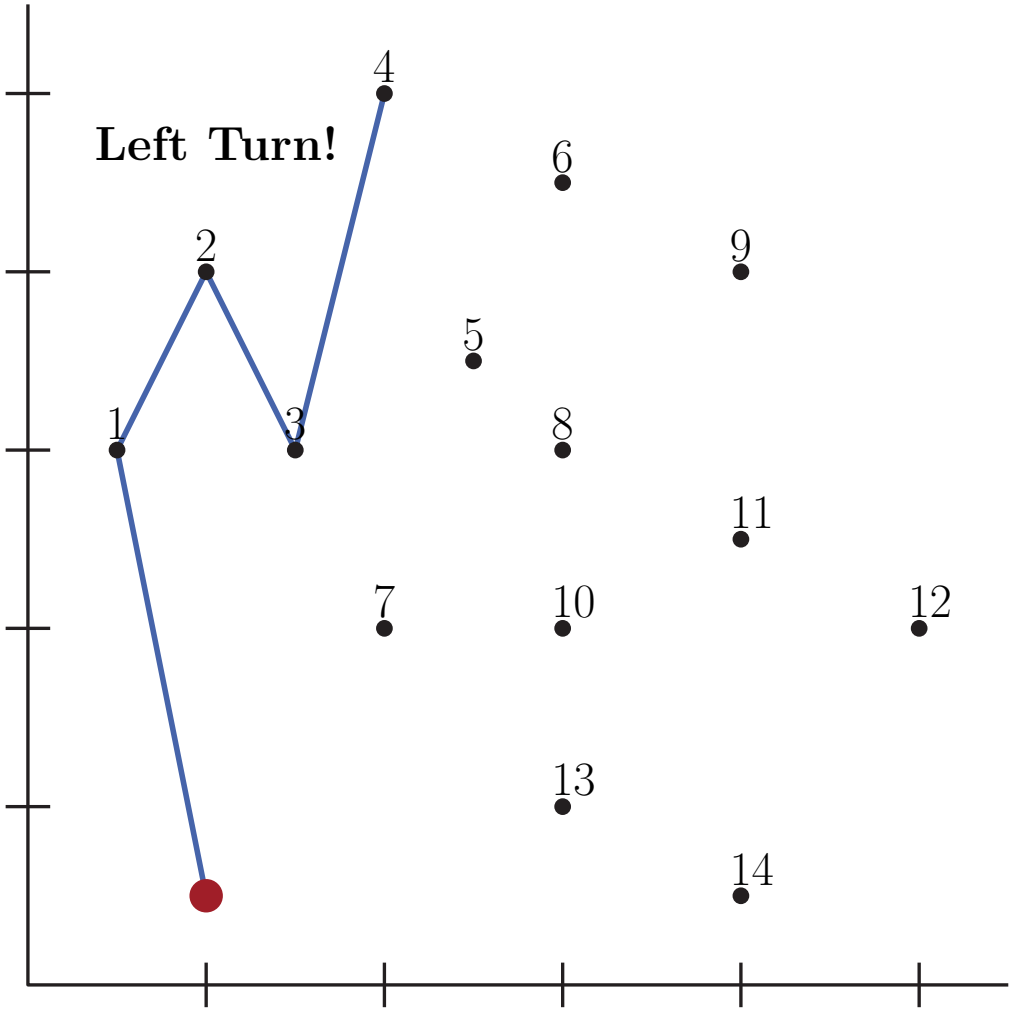
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

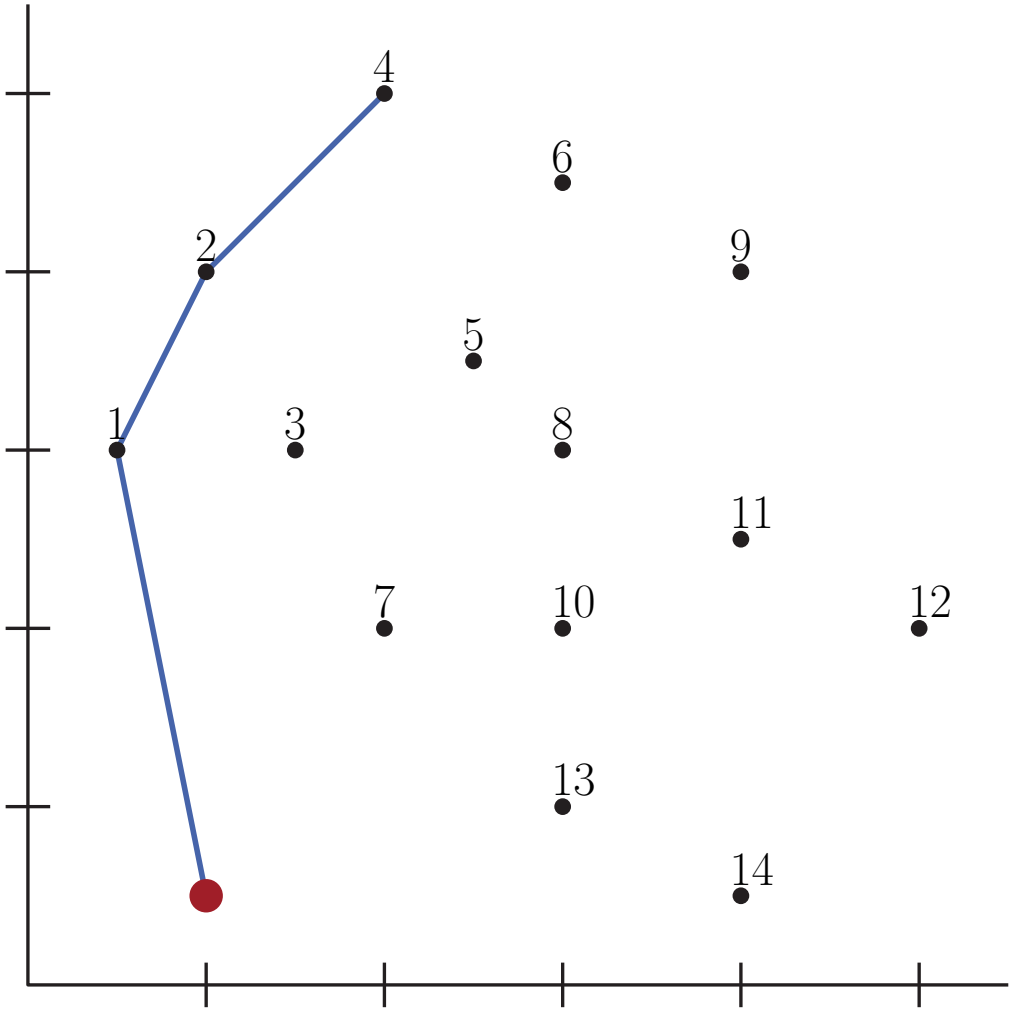
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

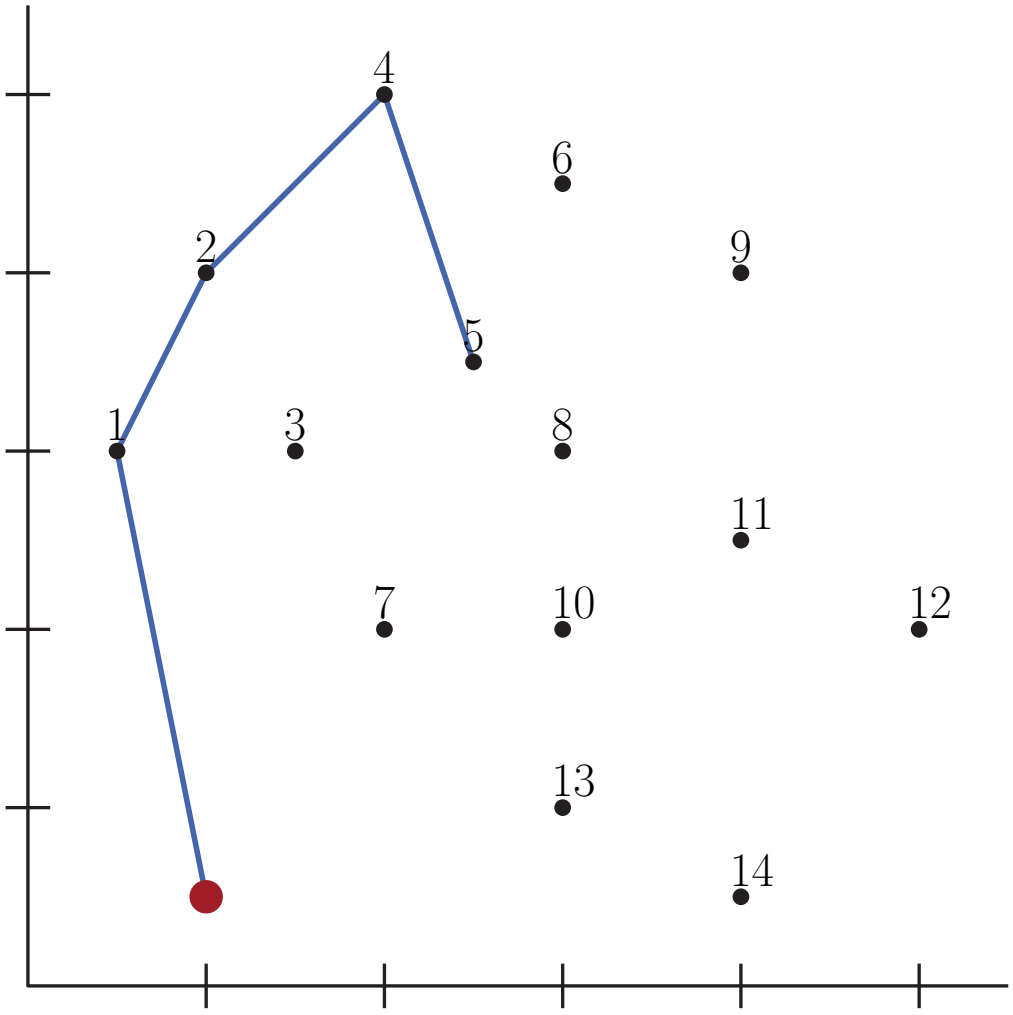
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

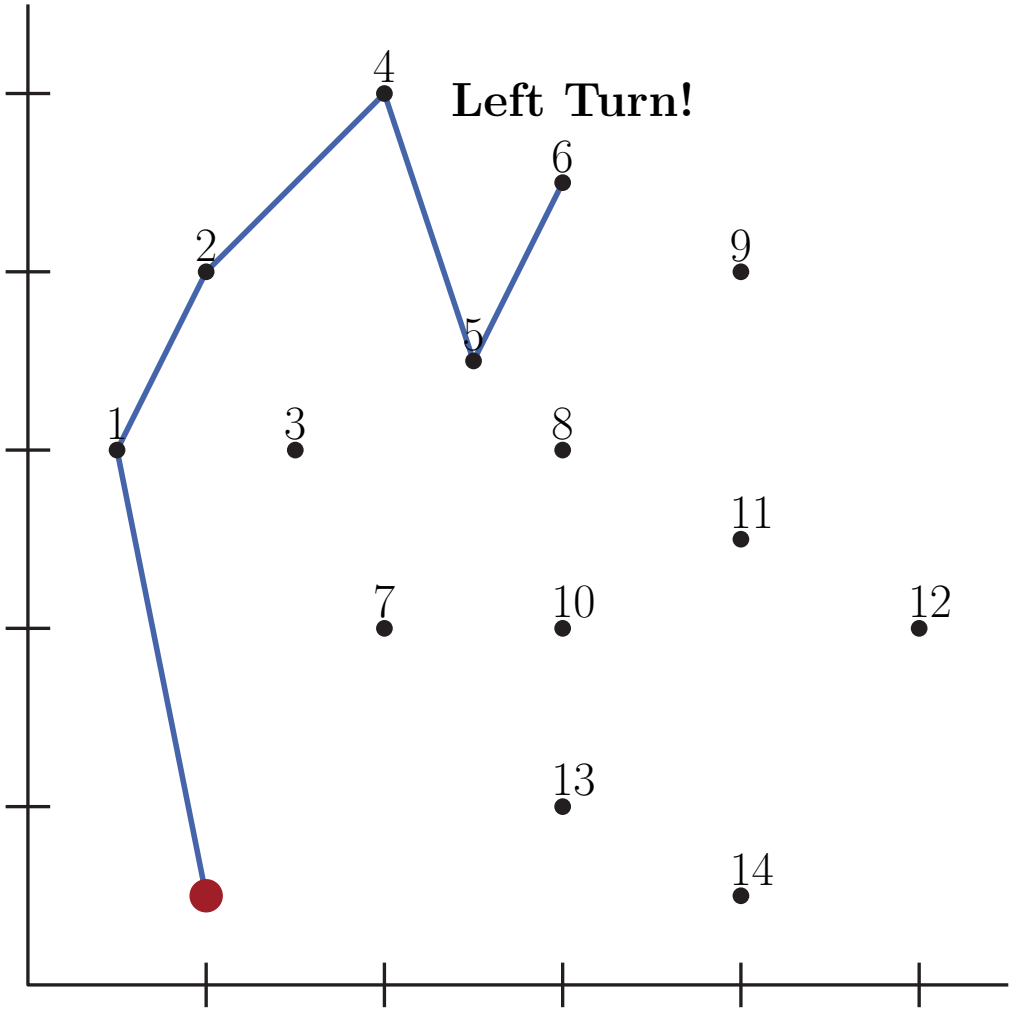
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

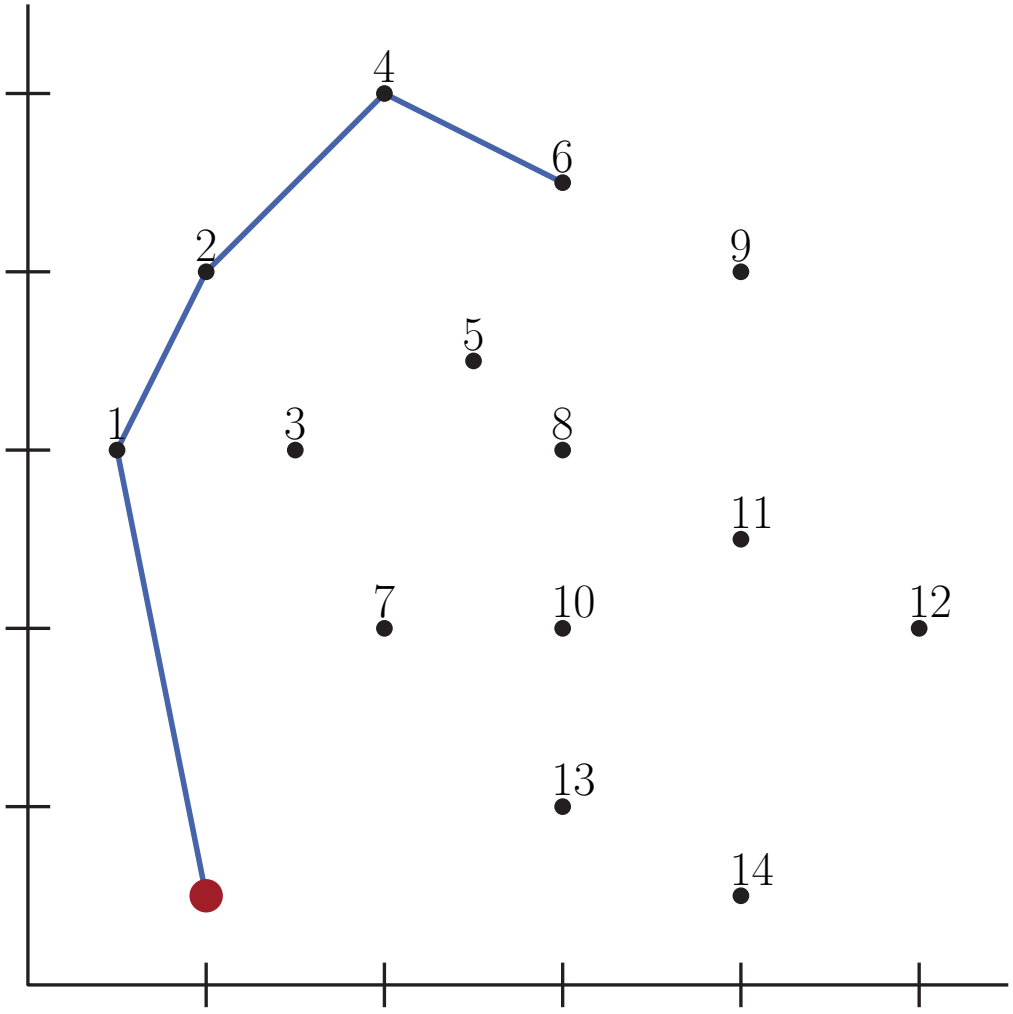
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

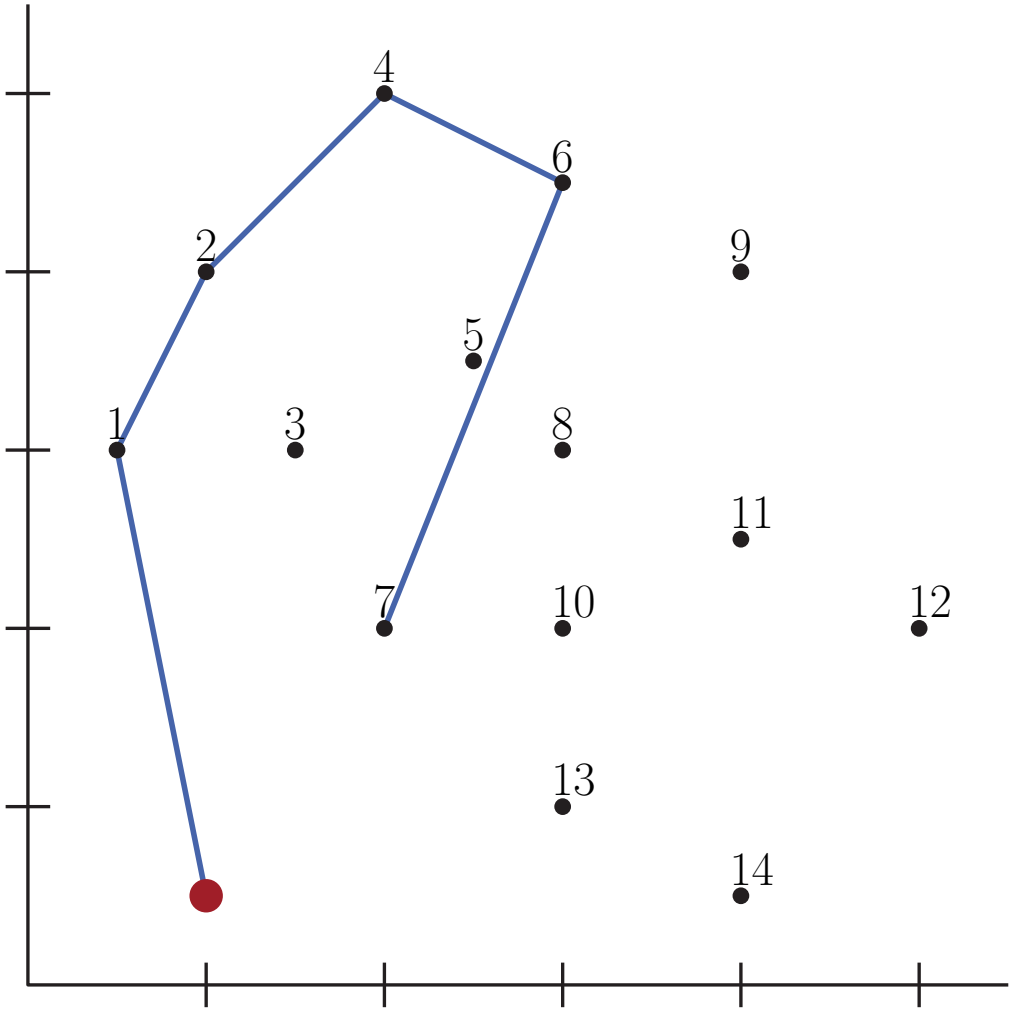
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

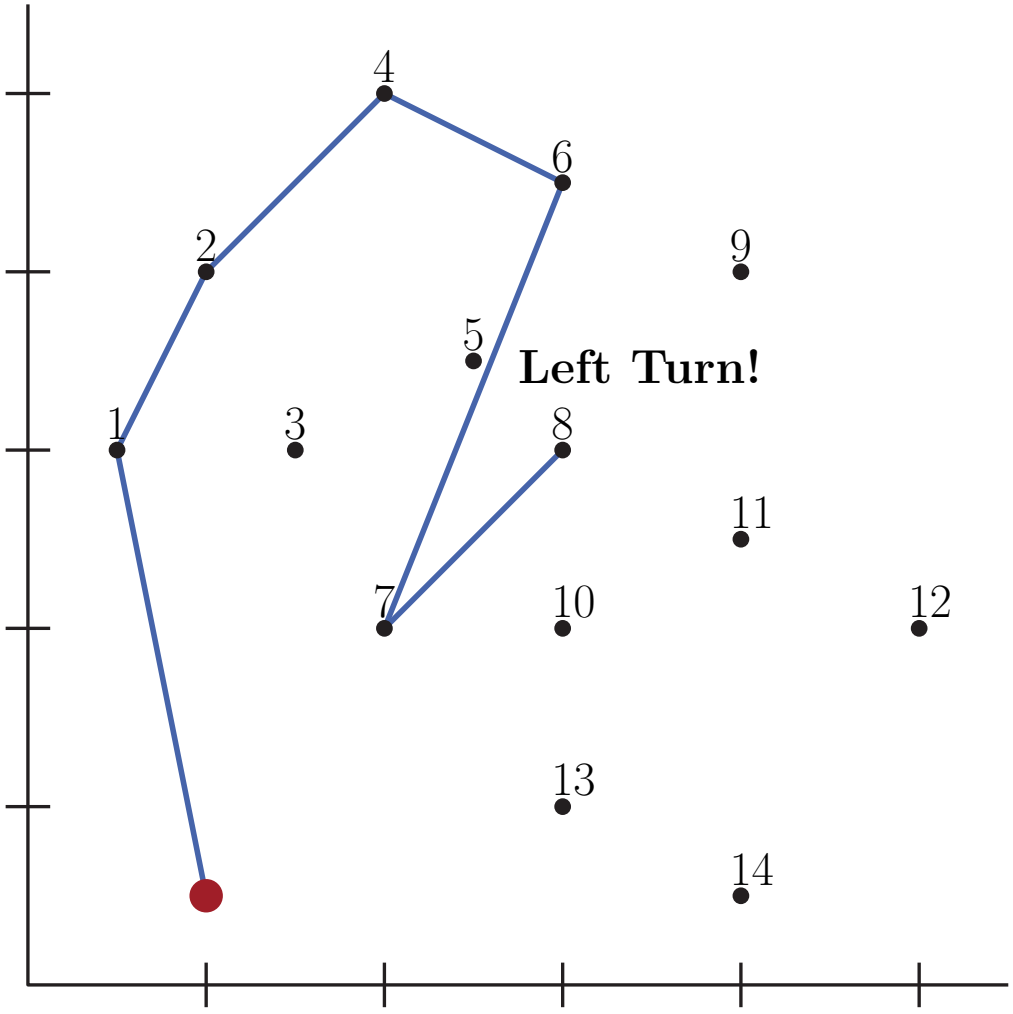
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

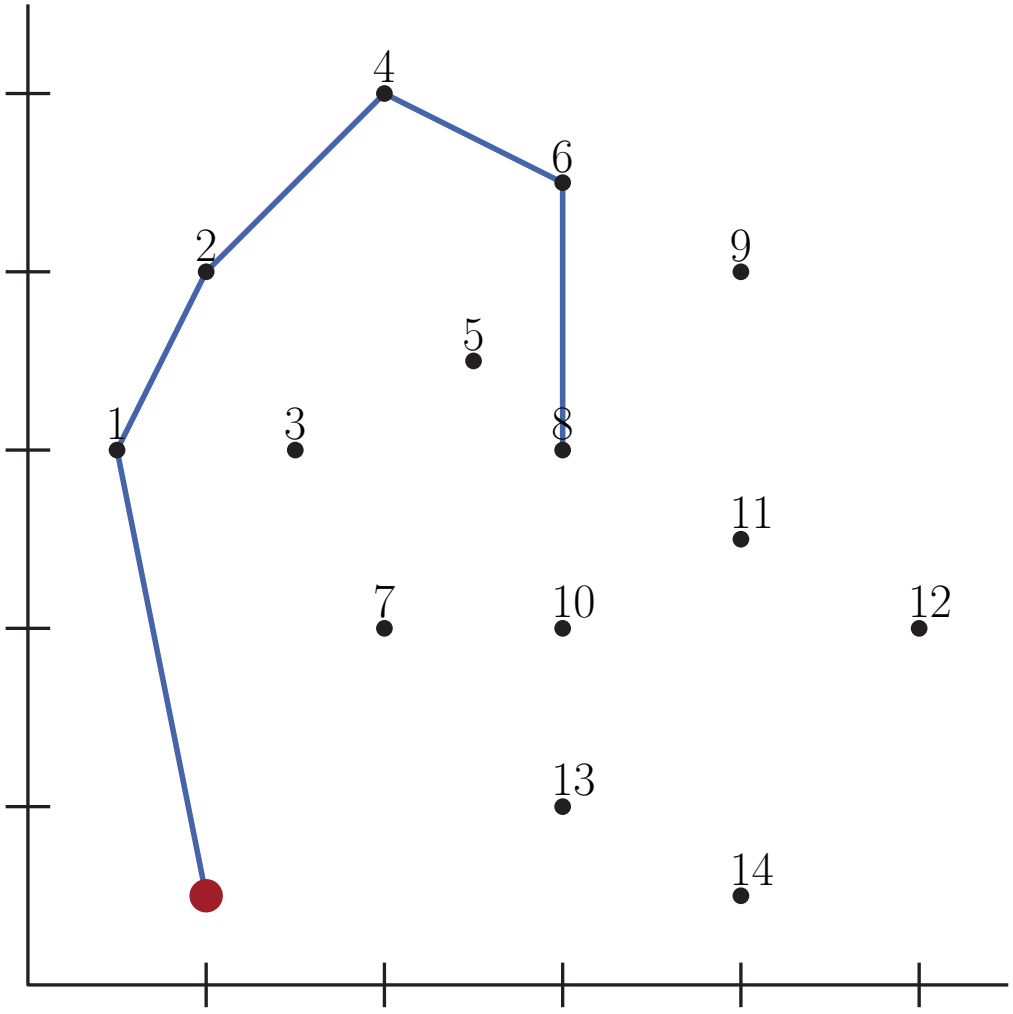
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

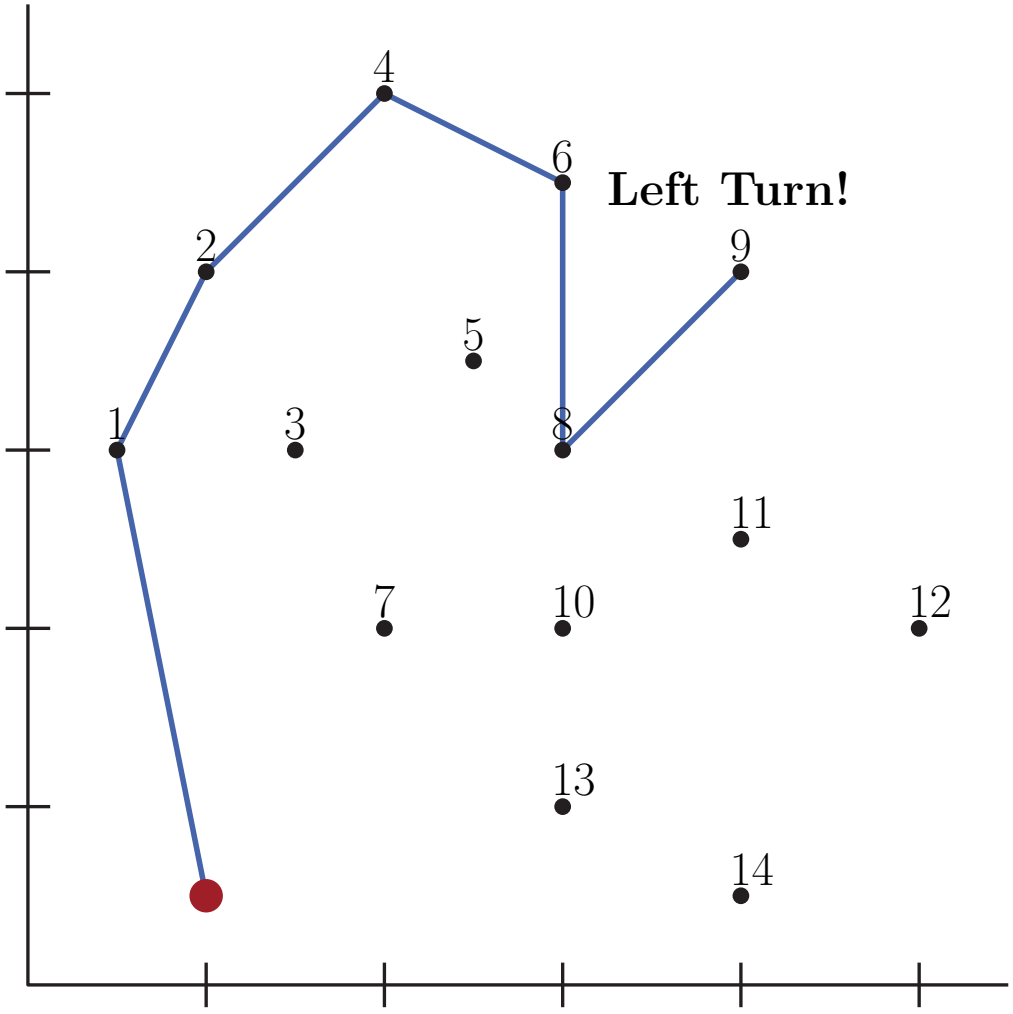
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

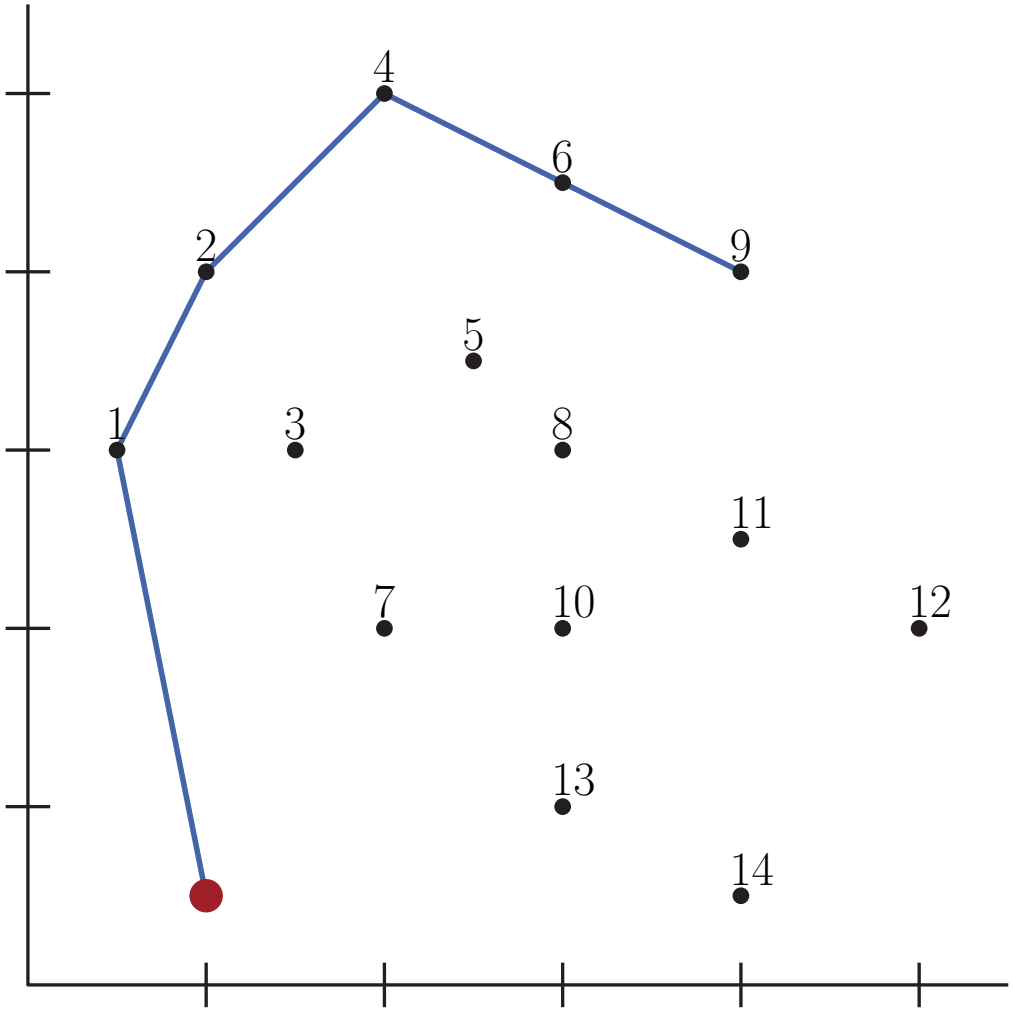
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

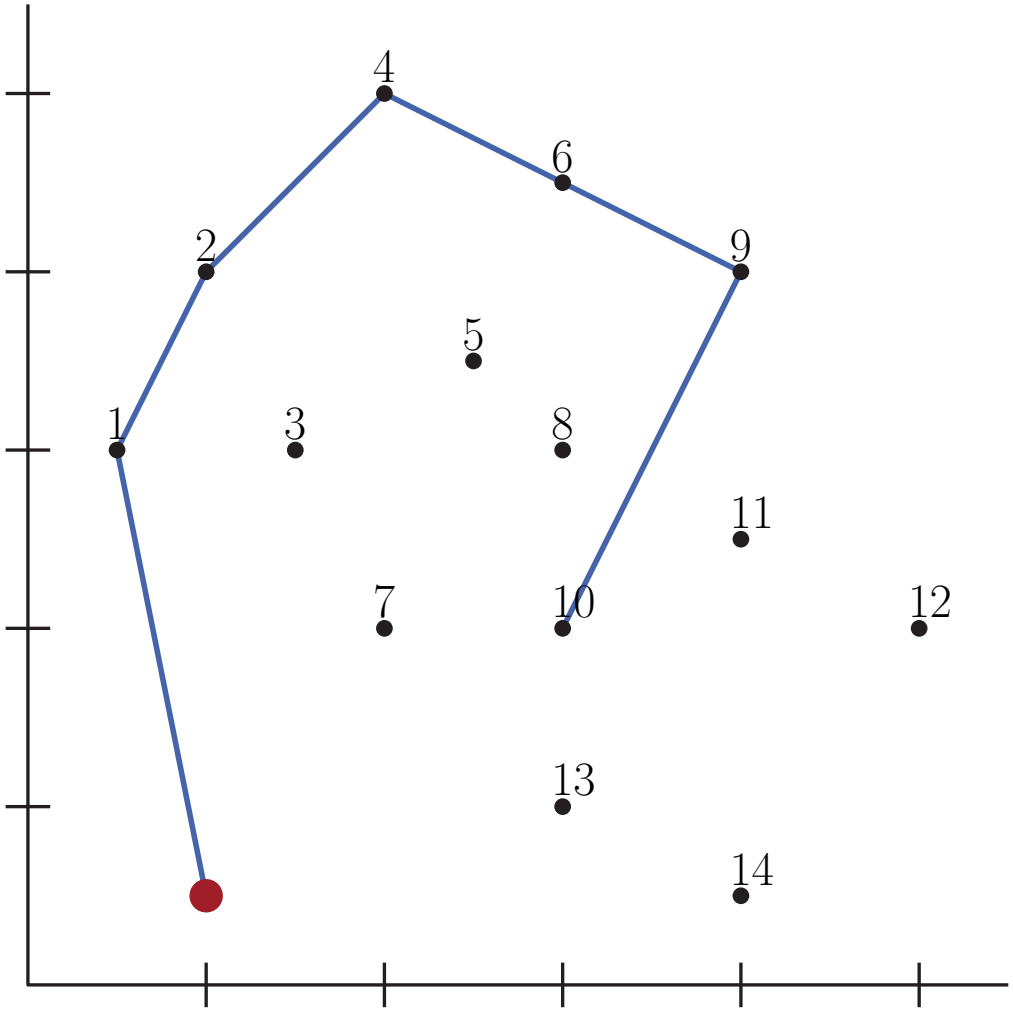
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

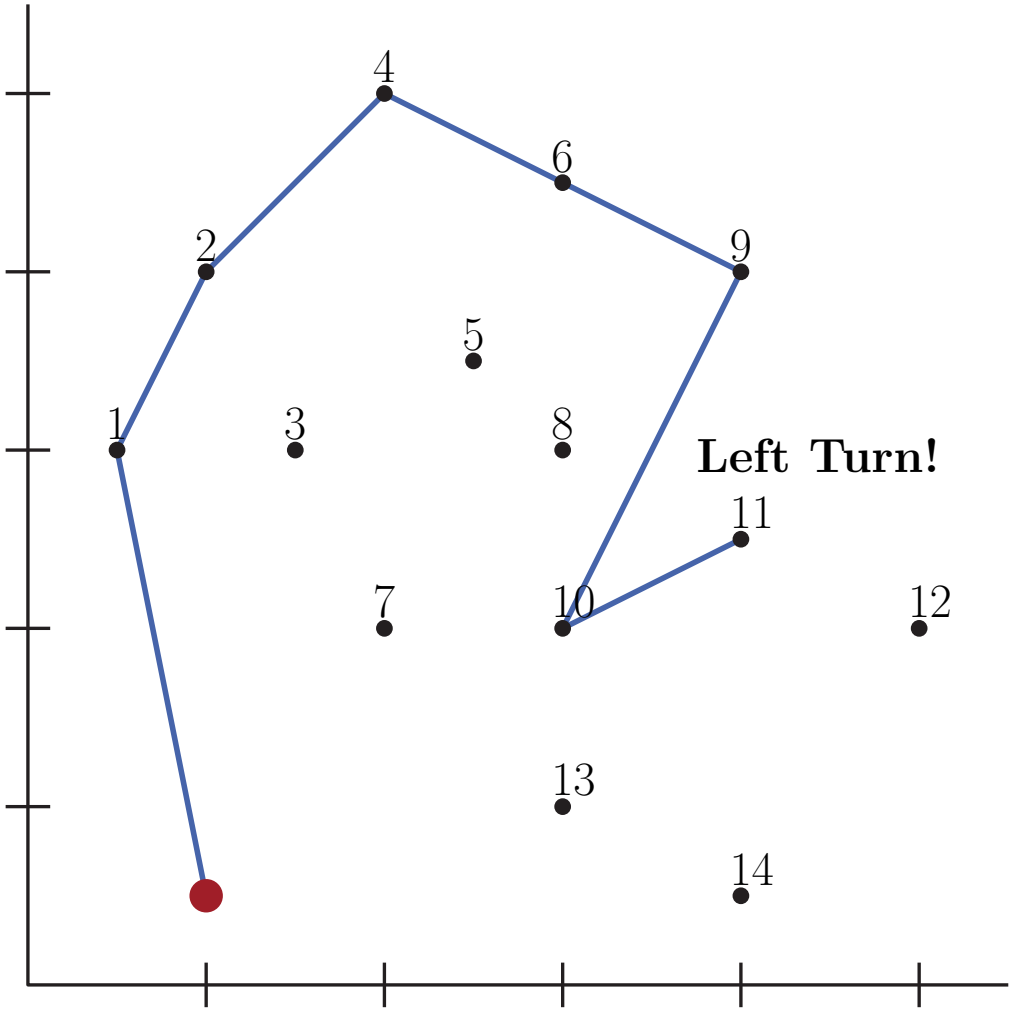
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

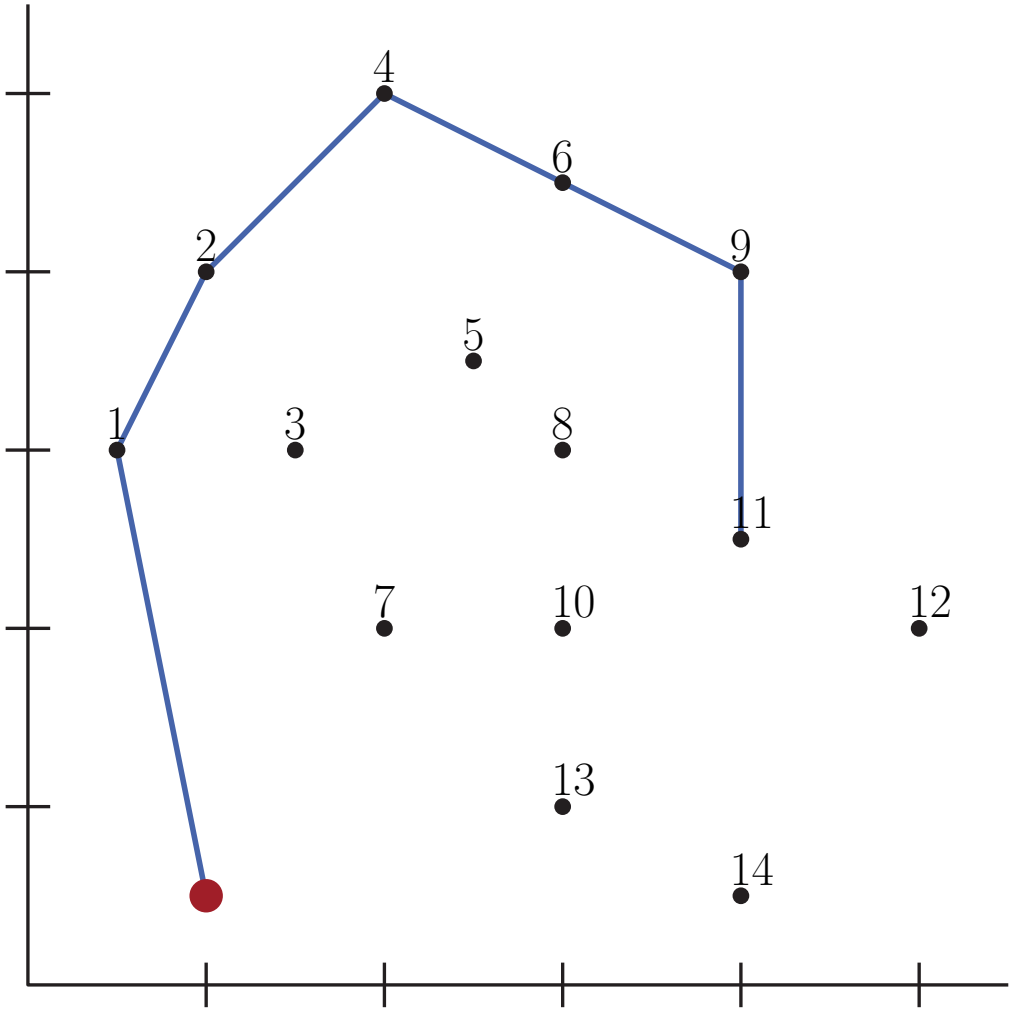
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

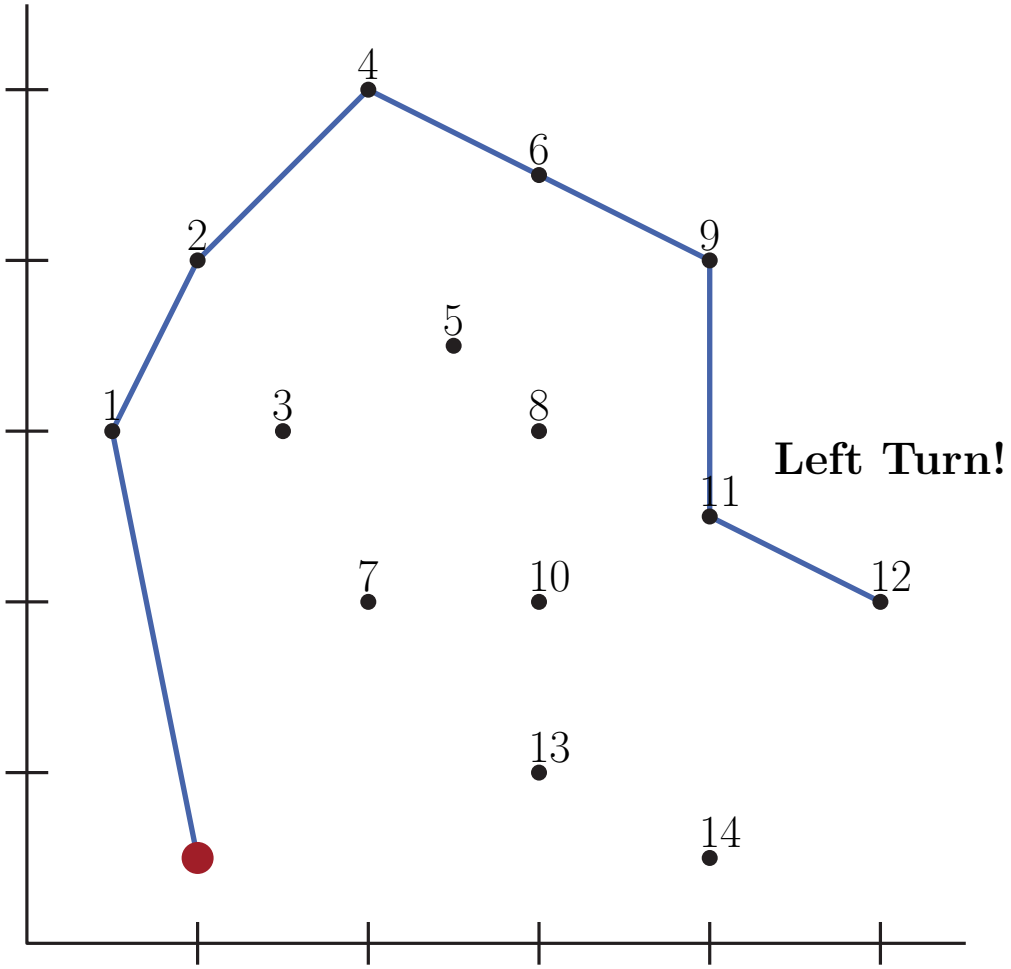
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

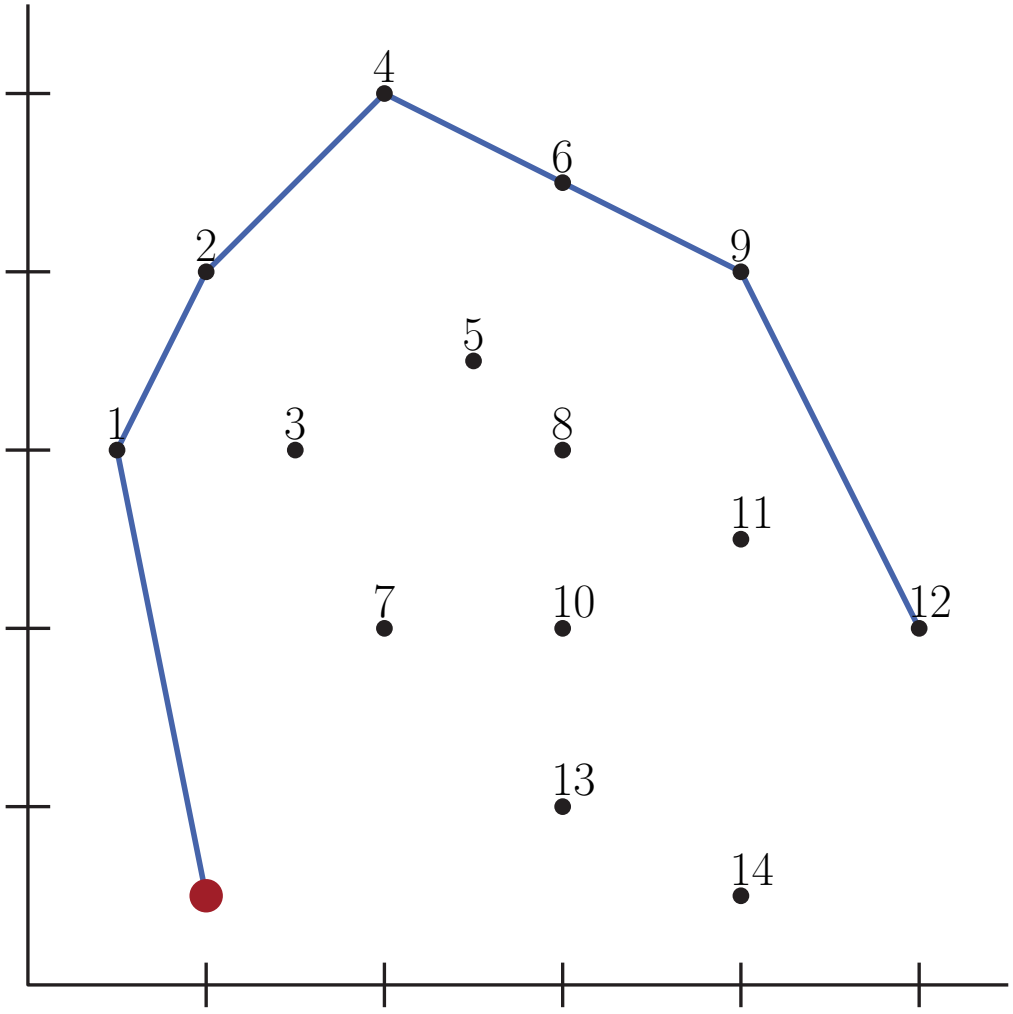
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

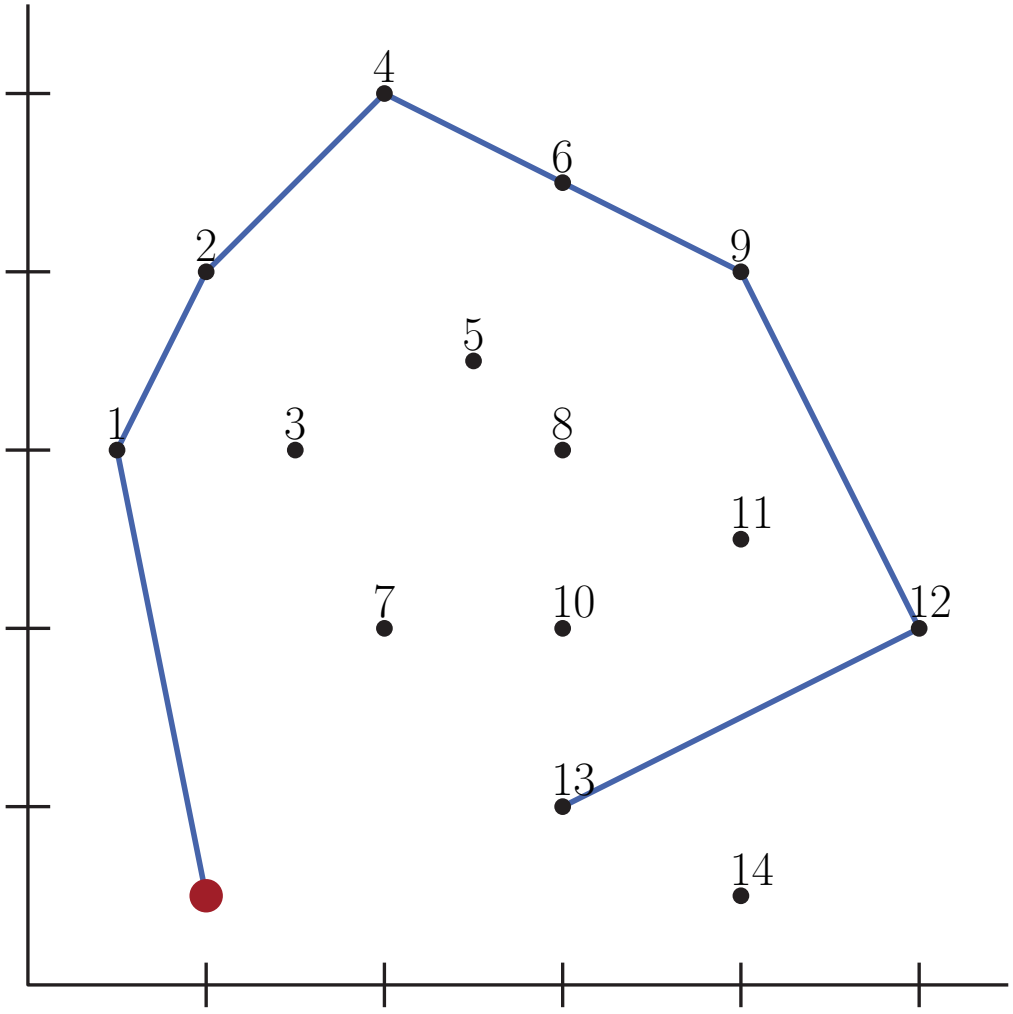
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

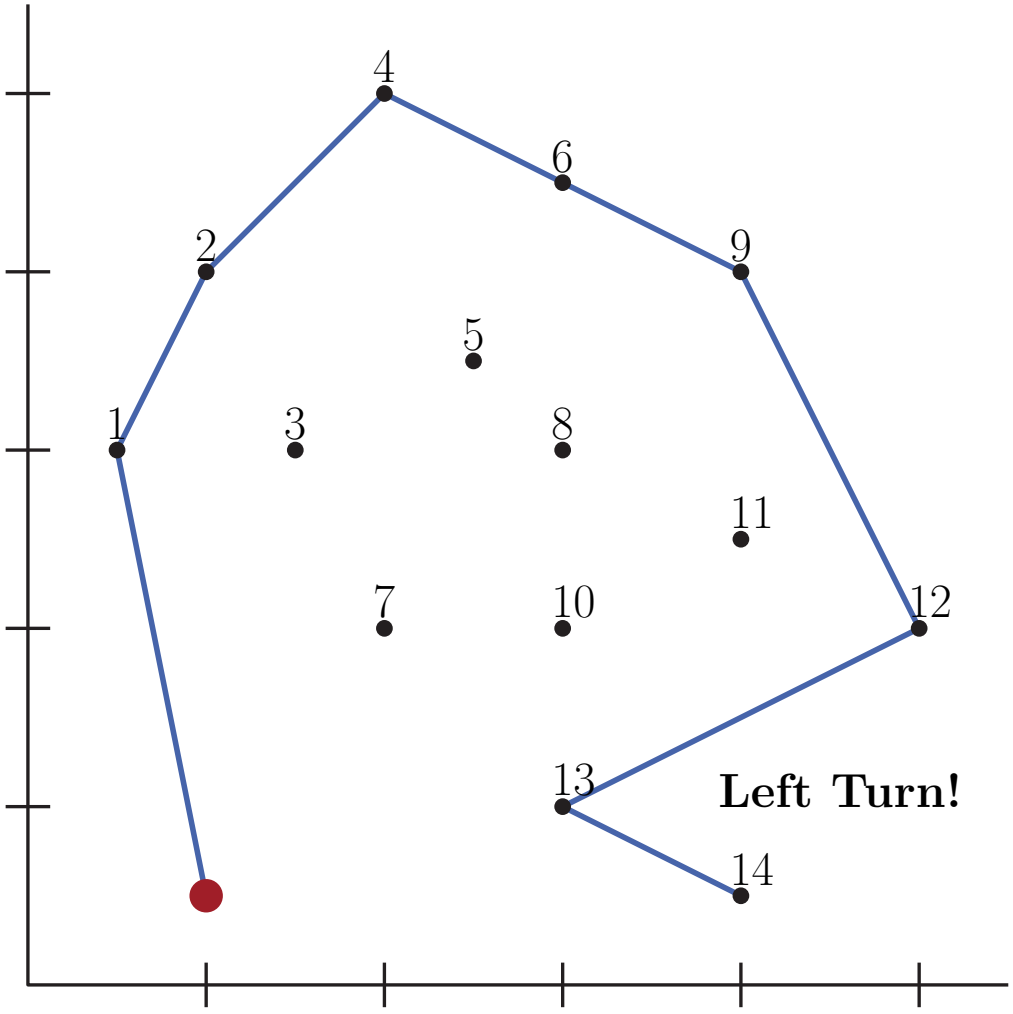
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

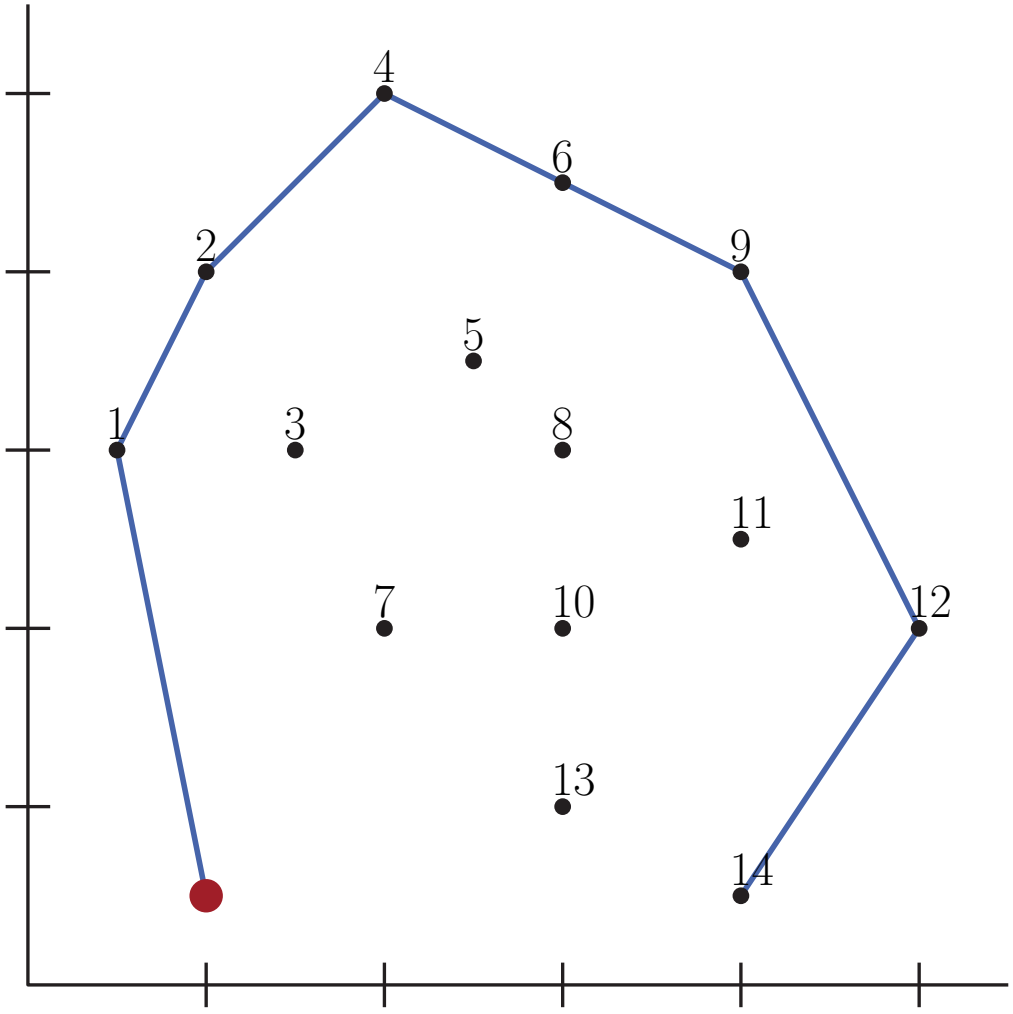
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

Graham-Scan

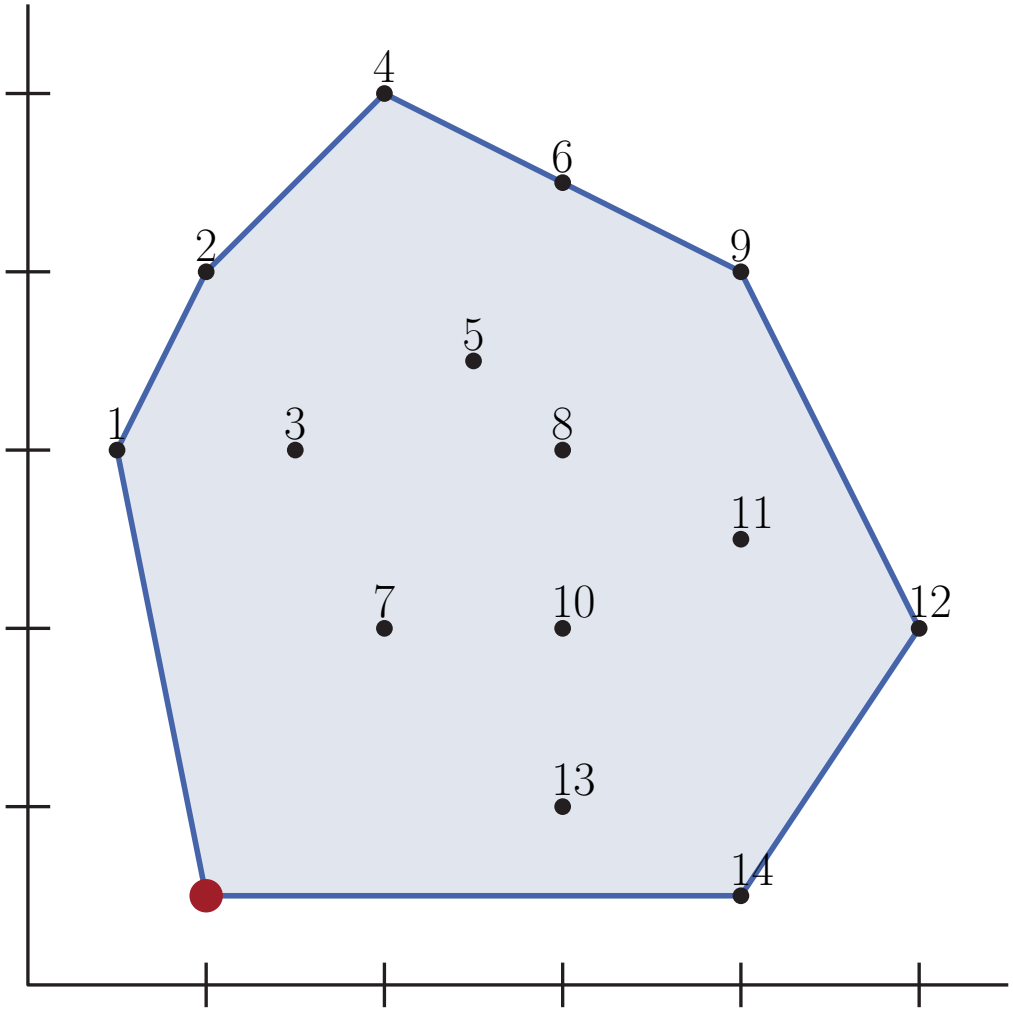
1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



Konvexe Hülle

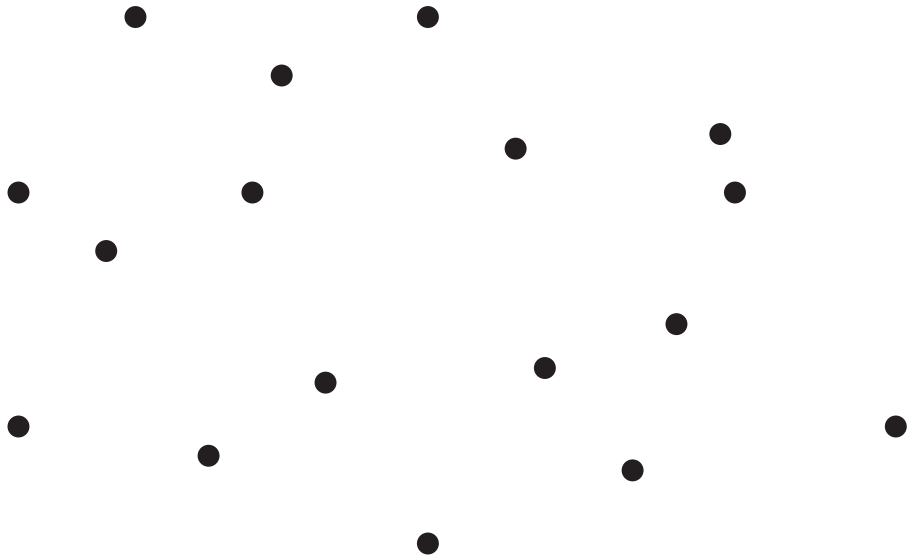
Graham-Scan

1. Finde Punkt P_0 mit kleinster y-Koordinate
   - Gibt es mehrere Punkte mit gleicher y-Koordinate, dann nehme Punkt mit kleinster x-Koordinate
2. Sortiere alle Punkte in absteigendem Winkel relativ zu P_0
3. Itereriere über alle Punkte P_i ($i > 2$) und betrachte Dreieck $H_{k-1} H_k P_i$ (wobei H_i i-ter Punkt in der aktuellen konvexen Hülle)
   - *Rechtsknick*: Füge P_i zur konvexen Hülle H hinzu
   - *Linksknick*: Lösche H_k aus bisheriger konvexen Hülle H



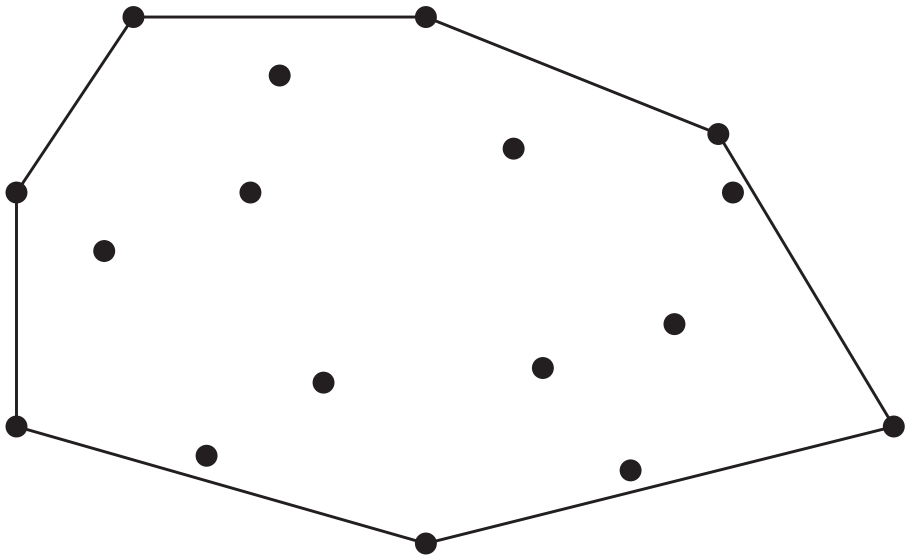
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



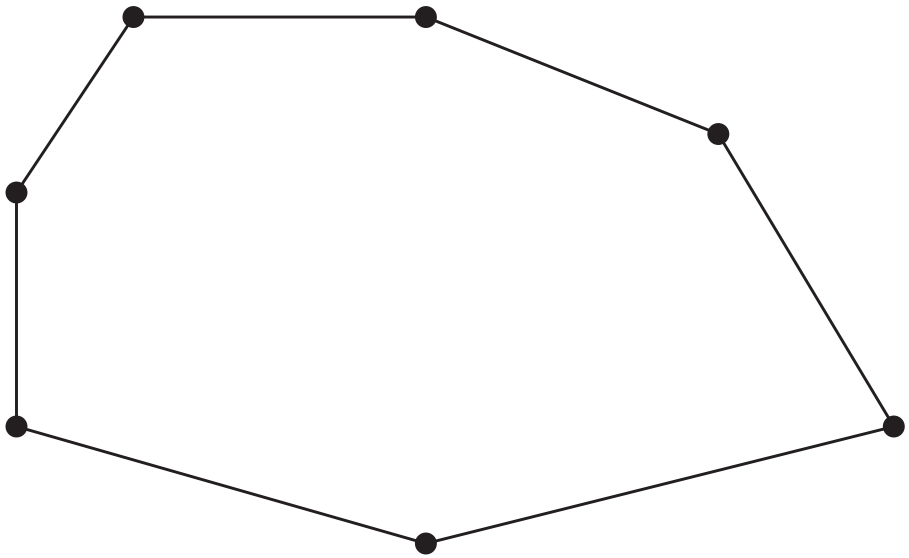
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



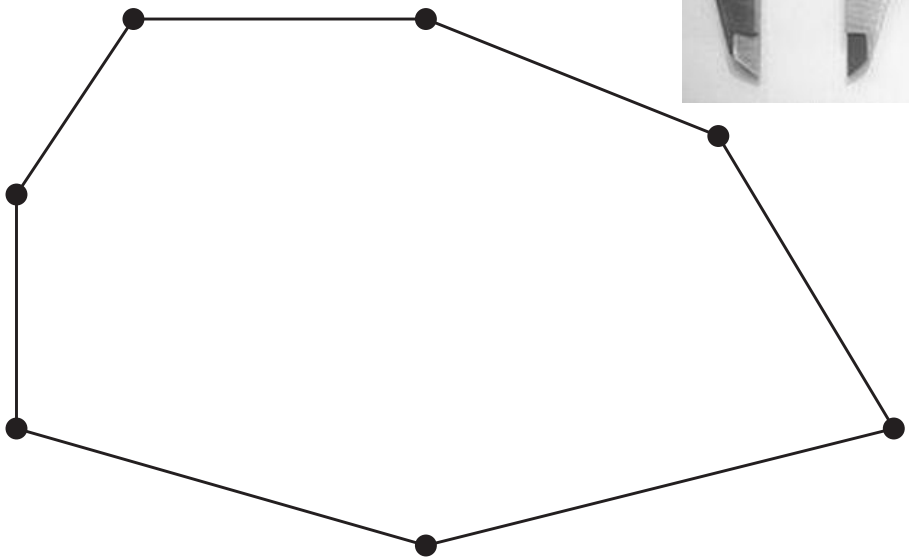
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



Konvexe Hülle

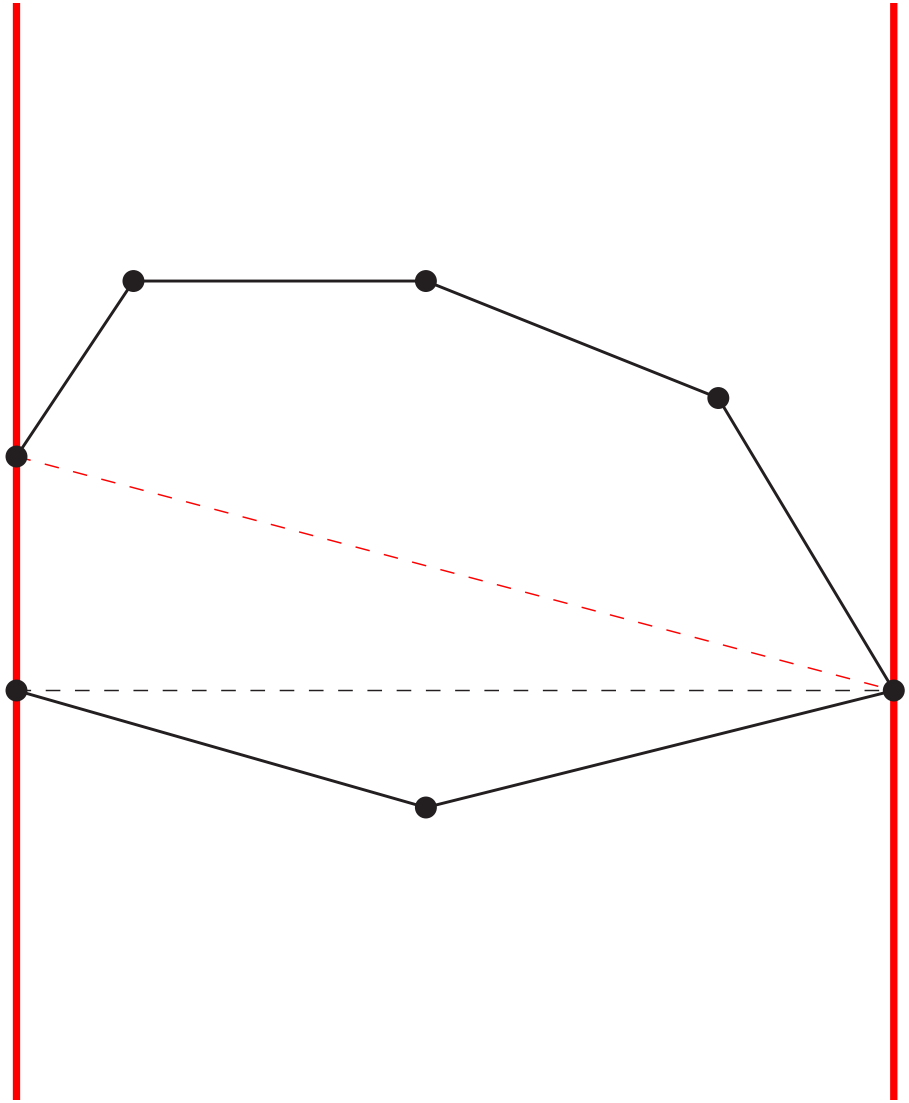
Anwendung: Durchmesser von Punktmenge: Rotating Calipers



Bildquelle: https://commons.wikimedia.org/wiki/File:3schieb.jpg

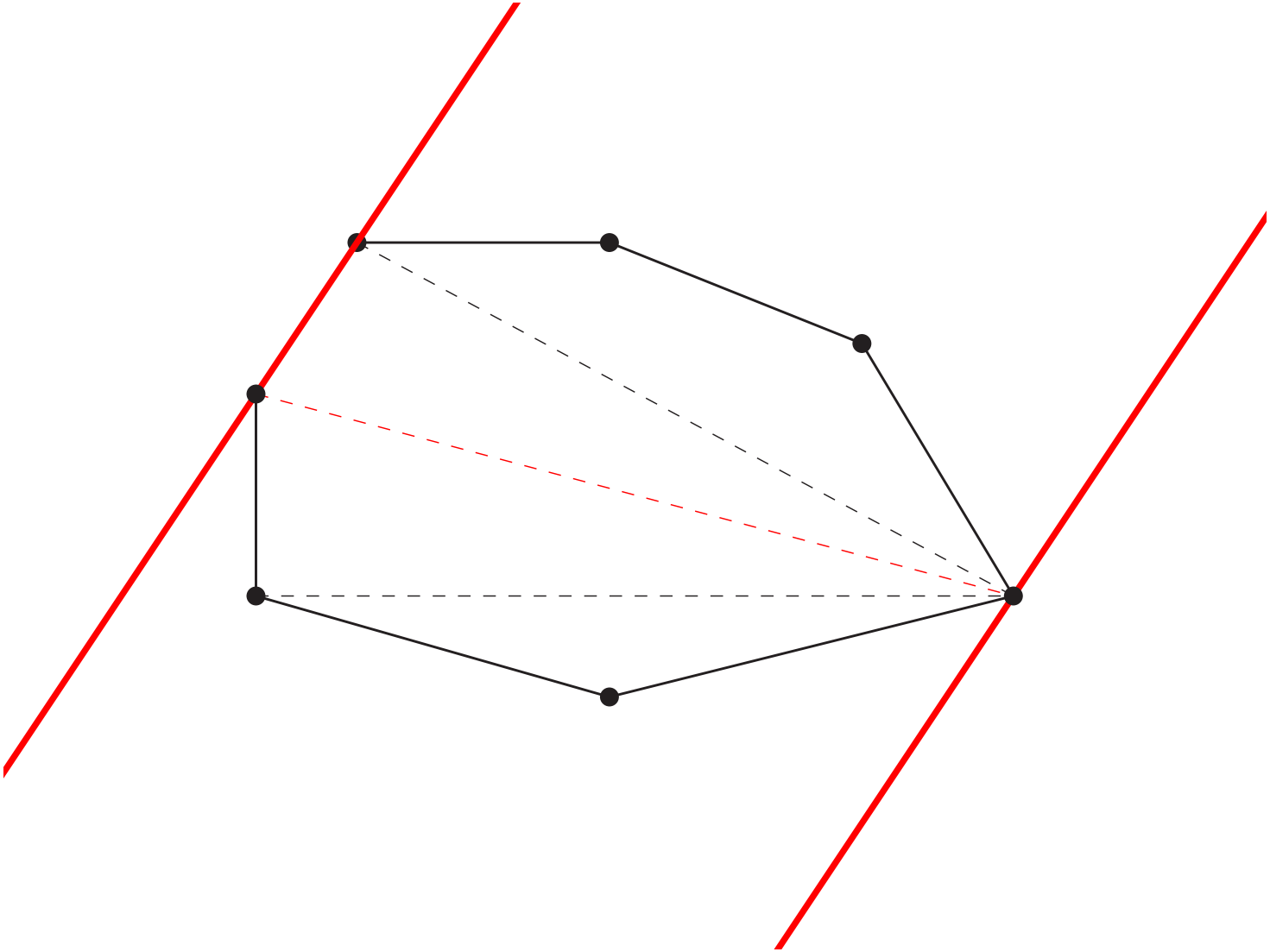
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



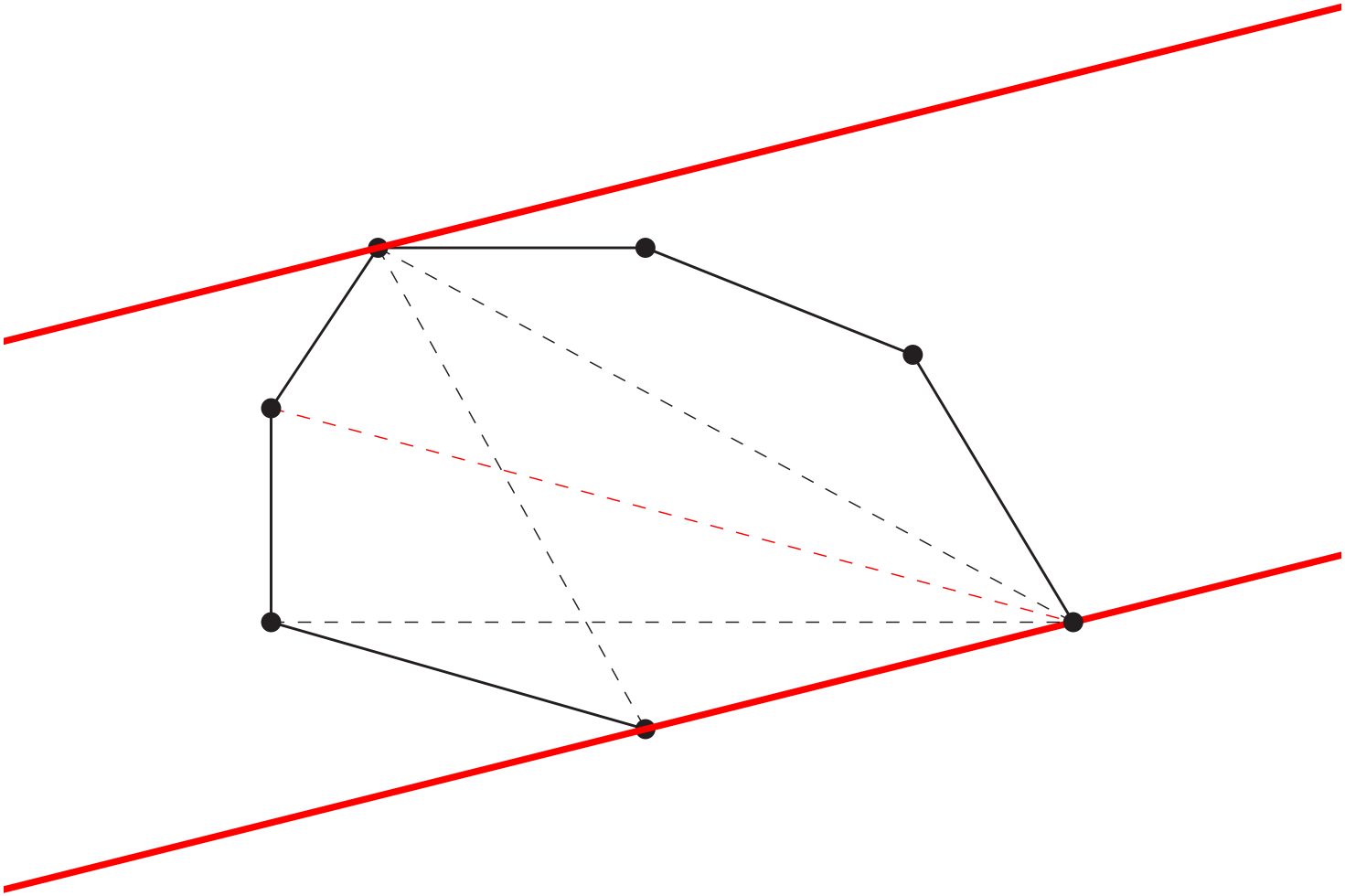
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



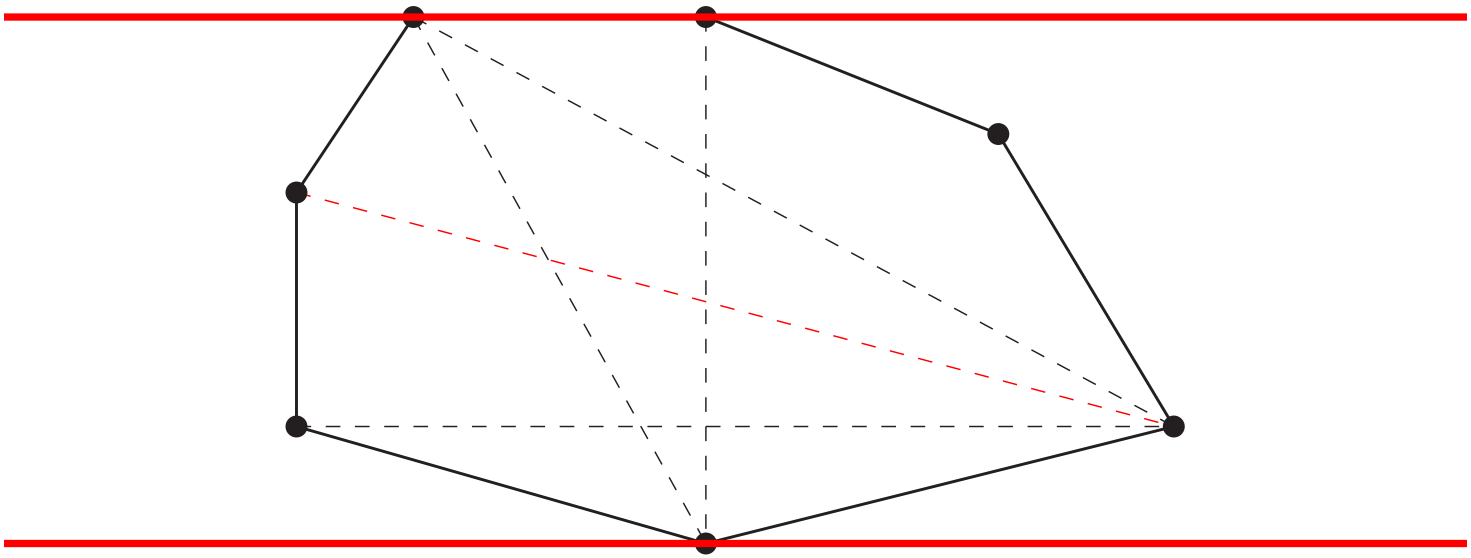
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



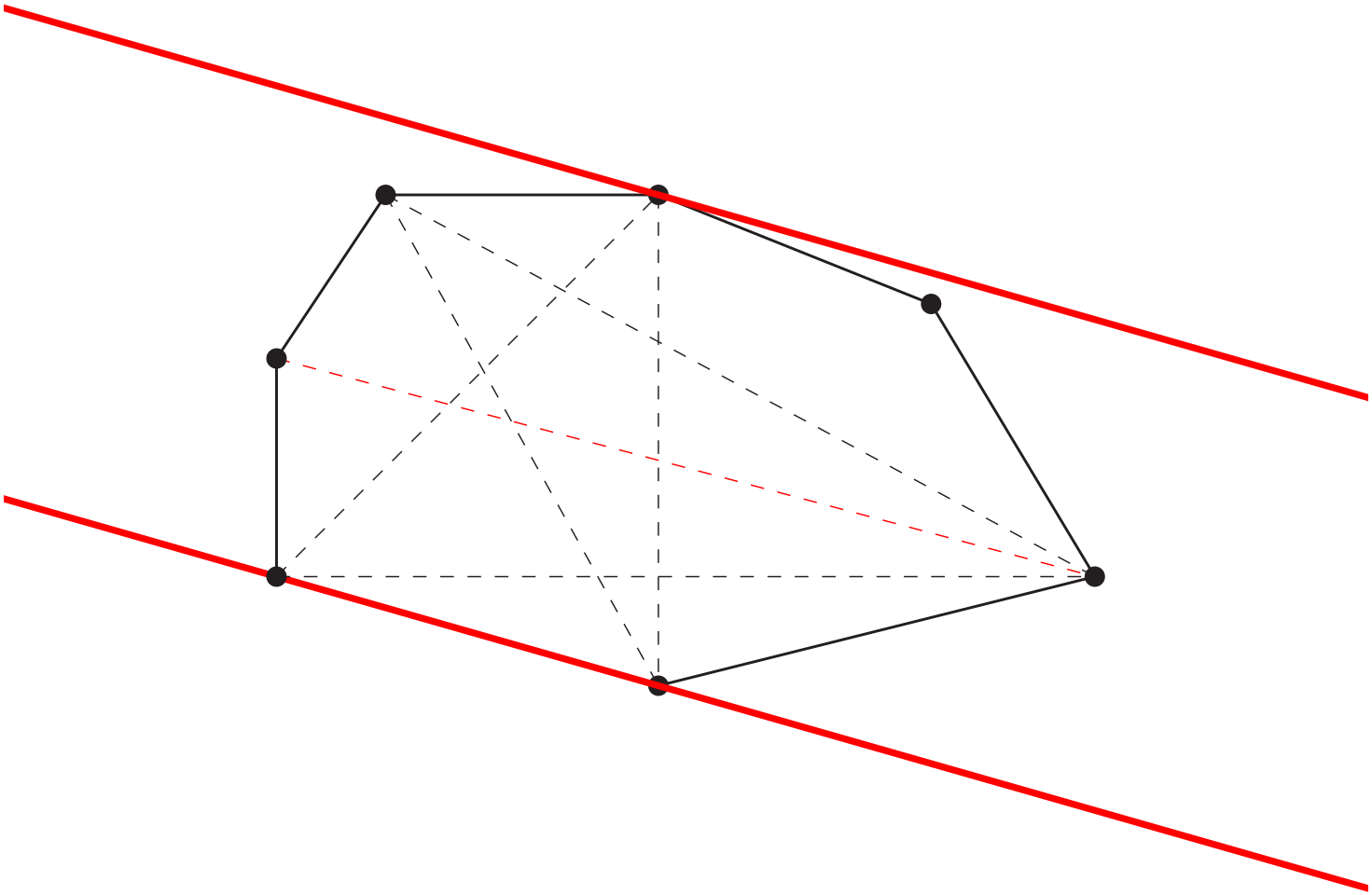
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



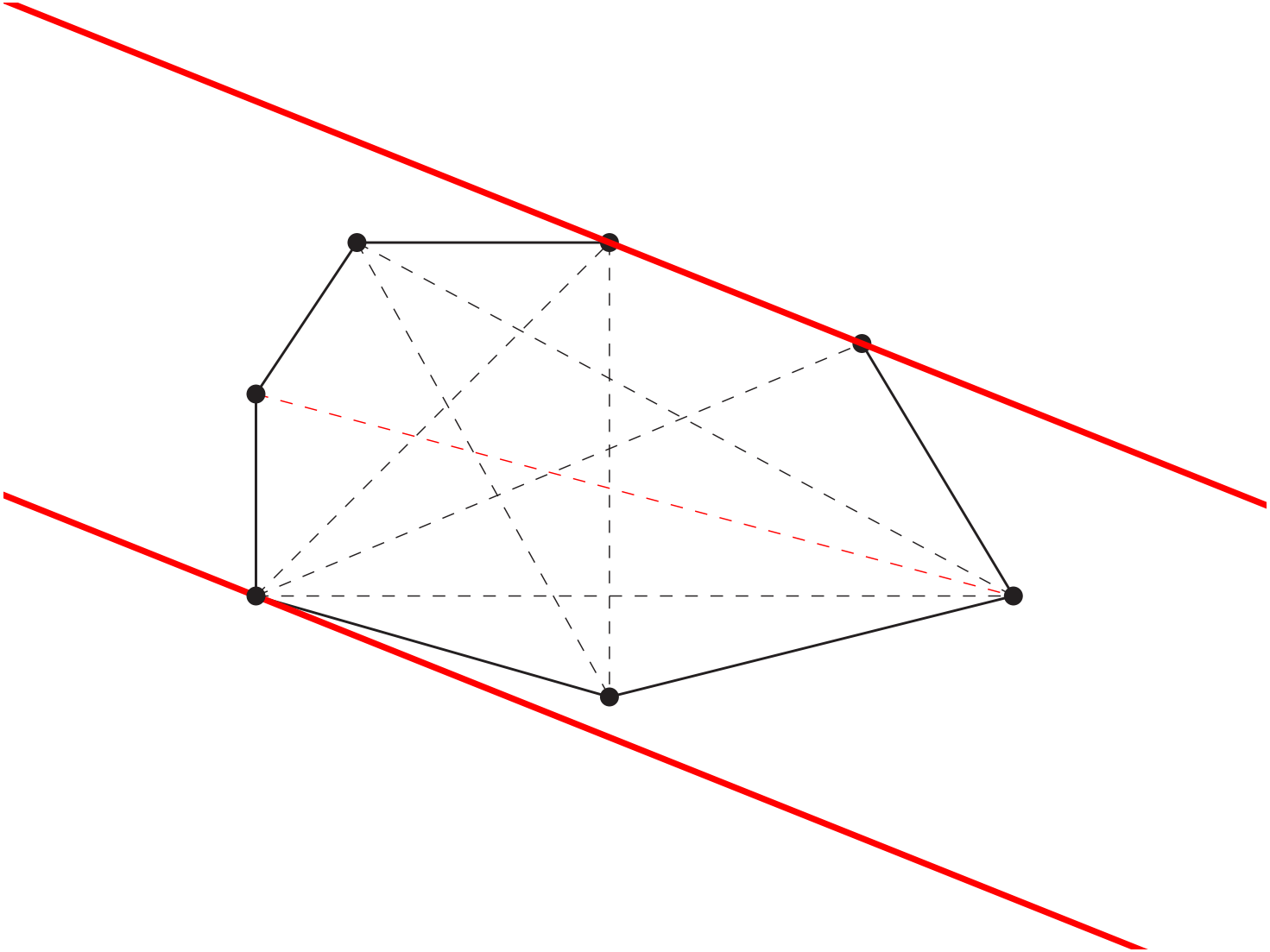
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



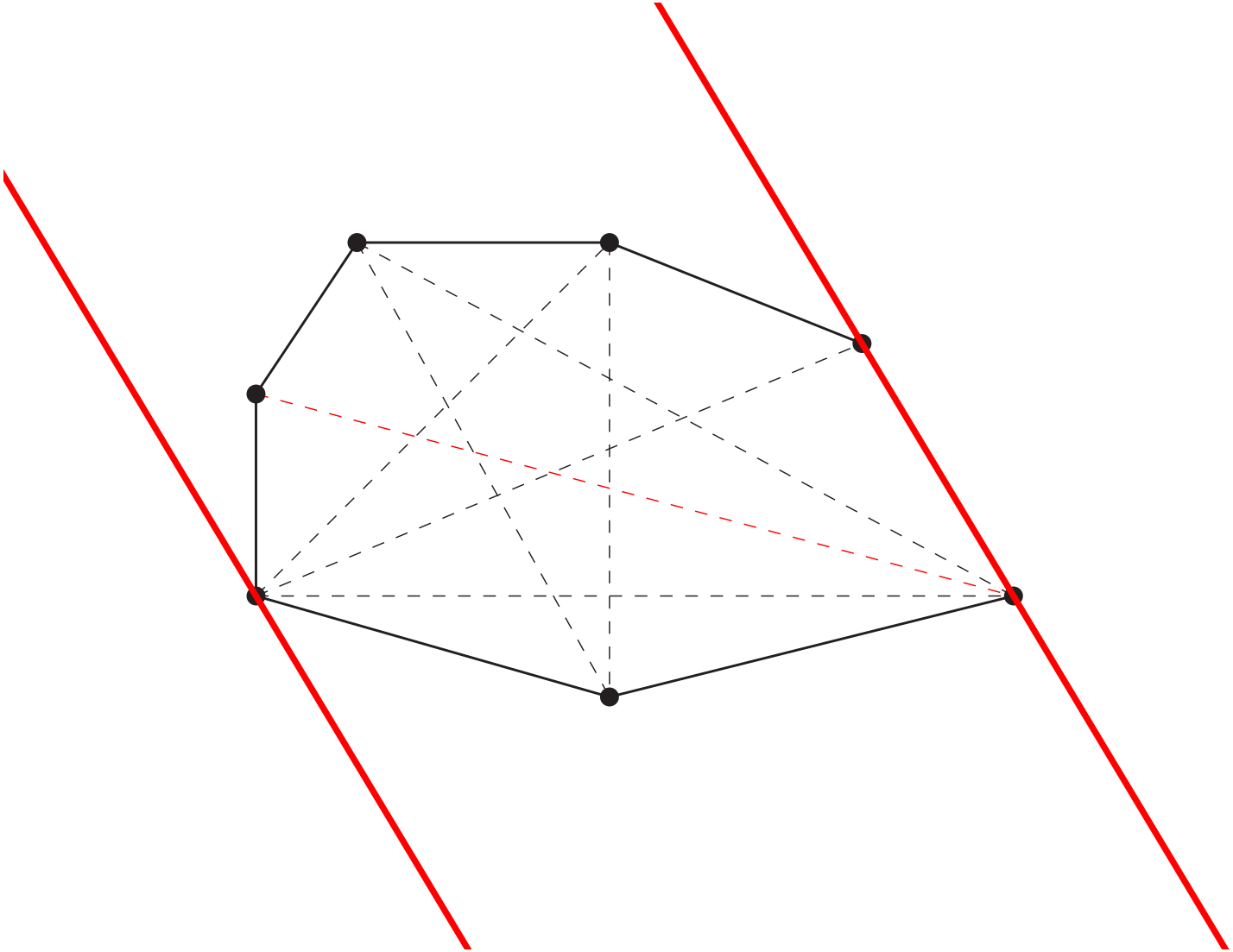
Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



Konvexe Hülle

Anwendung: Durchmesser von Punktmenge: Rotating Calipers



Ende!

Feierabend!