Algorithmentechnik — Übung 2

http://i11www.ira.uka.de/teaching/WS_0505/algotech

Robert Görke (rgoerke@ira.uka.de)

WS 0506

Übersicht

Union Find
- Vorbereitung
- Laufzeit

Erst Union dann Find (Problem 2)

Postamt-Platzierungs-Problem (Problem 1)

Graustufenbilder (Problem 4)

Ende

Rang

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)
- Für alle Knoten v: $\mathrm{rank}(v) < \mathrm{rank}(p[v])$

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)
- Für alle Knoten v: $\mathrm{rank}(v) < \mathrm{rank}(p[v])$
- $\mathrm{rank}_{\mathrm{init}}(v) = 0, \quad \mathrm{rank}(v) \nearrow$ monoton steigend

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)
- Für alle Knoten v: $\mathrm{rank}(v) < \mathrm{rank}(p[v])$
- $\mathrm{rank}_{\mathrm{init}}(v) = 0, \quad \mathrm{rank}(v) \nearrow$ monoton steigend
- Ab Zeitpunkt wo $v \neq$ Wurzel: $\mathrm{rank}(v) =$ konstant

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)
- Für alle Knoten v: $\mathrm{rank}(v) < \mathrm{rank}(p[v])$
- $\mathrm{rank}_{\mathrm{init}}(v) = 0, \quad \mathrm{rank}(v) \nearrow$ monoton steigend
- Ab Zeitpunkt wo $v \neq$ Wurzel: $\mathrm{rank}(v) =$ konstant
- Auch $\mathrm{rank}(p[v]) \nearrow$ monoton steigend

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)
- Für alle Knoten v: $\mathrm{rank}(v) < \mathrm{rank}(p[v])$
- $\mathrm{rank}_{\mathrm{init}}(v) = 0, \quad \mathrm{rank}(v) \nearrow$ monoton steigend
- Ab Zeitpunkt wo $v \neq$ Wurzel: $\mathrm{rank}(v) =$ konstant
- Auch $\mathrm{rank}(p[v]) \nearrow$ monoton steigend
- Für alle Wurzeln x: $|T(x)| \geq 2^{\mathrm{rank}(x)}$ (Induktion)

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)
- Für alle Knoten v: $\mathrm{rank}(v) < \mathrm{rank}(p[v])$
- $\mathrm{rank}_{\mathrm{init}}(v) = 0, \quad \mathrm{rank}(v) \nearrow$ monoton steigend
- Ab Zeitpunkt wo $v \neq$ Wurzel: $\mathrm{rank}(v) =$ konstant
- Auch $\mathrm{rank}(p[v]) \nearrow$ monoton steigend
- Für alle Wurzeln x: $|T(x)| \geq 2^{\mathrm{rank}(x)}$ (Induktion)
- $|$ Knoten mit Rang $r| \leq \frac{n}{2^r}$ (immer!) (Induktion)

Rang

- $\mathrm{rank}(v) = |$ längster Pfad zu Blatt in $T(v)|$ (jemals)
- Für alle Knoten v: $\mathrm{rank}(v) < \mathrm{rank}(p[v])$
- $\mathrm{rank}_{\mathrm{init}}(v) = 0, \quad \mathrm{rank}(v) \nearrow$ monoton steigend
- Ab Zeitpunkt wo $v \neq$ Wurzel: $\mathrm{rank}(v) =$ konstant
- Auch $\mathrm{rank}(p[v]) \nearrow$ monoton steigend
- Für alle Wurzeln x: $|T(x)| \geq 2^{\mathrm{rank}(x)}$ (Induktion)
- $|$ Knoten mit Rang $r| \leq \frac{n}{2^r}$ (immer!) (Induktion)
- $\forall v : \mathrm{rank}(v) \leq \lfloor \log n \rfloor$ (Kor.)

Ranggruppen

$$\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\} \qquad j \geq 1$$

Ranggruppen

$$\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\} \qquad j \geq 1$$

$$\gamma_j := \begin{cases} \emptyset & \text{wenn } \log^{(j)} n \text{ undefiniert,} \\ \{v : r(v) = 0\} & \text{falls } \lfloor \log^{(j)} n \rfloor = 0. \end{cases}$$

Ranggruppen

$$\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\} \qquad j \geq 1$$

$$\gamma_j := \begin{cases} \emptyset & \text{wenn } \log^{(j)} n \text{ undefiniert,} \\ \{v : r(v) = 0\} & \text{falls } \lfloor \log^{(j)} n \rfloor = 0. \end{cases}$$

$$\begin{aligned}
\gamma_1 &= \{v : \log^{(2)} 66000 < r(v) \leq \log 66000\} = \{v : 4 < r(v) \leq 16\} \\
\gamma_2 &= \{v : \log^{(3)} 66000 < r(v) \leq \log^{(2)} 66000\} = \{v : 2 < r(v) \leq 4\} \\
\gamma_3 &= \{v : \log^{(4)} 66000 < r(v) \leq \log^{(3)} 66000\} = \{v : 1 < r(v) \leq 2\} \\
\gamma_4 &= \{v : \log^{(5)} 66000 < r(v) \leq \log^{(4)} 66000\} = \{v : 0 < r(v) \leq 1\} \\
\gamma_5 &= \{v : \log^{(6)} 66000 < r(v) \leq \log^{(5)} 66000\} = \{v : \log 0 < r(v) \leq 0\} \\
&= \{v : r(v) = 0\}
\end{aligned}$$

Ranggruppen

$$\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\} \qquad j \geq 1$$

$$\gamma_j := \begin{cases} \emptyset & \text{wenn } \log^{(j)} n \text{ undefiniert,} \\ \{v : r(v) = 0\} & \text{falls } \lfloor \log^{(j)} n \rfloor = 0. \end{cases}$$

$$\begin{aligned}
\gamma_1 &= \{v : \log^{(2)} 66000 < r(v) \leq \log 66000\} = \{v : 4 < r(v) \leq 16\} \\
\gamma_2 &= \{v : \log^{(3)} 66000 < r(v) \leq \log^{(2)} 66000\} = \{v : 2 < r(v) \leq 4\} \\
\gamma_3 &= \{v : \log^{(4)} 66000 < r(v) \leq \log^{(3)} 66000\} = \{v : 1 < r(v) \leq 2\} \\
\gamma_4 &= \{v : \log^{(5)} 66000 < r(v) \leq \log^{(4)} 66000\} = \{v : 0 < r(v) \leq 1\} \\
\gamma_5 &= \{v : \log^{(6)} 66000 < r(v) \leq \log^{(5)} 66000\} = \{v : \log 0 < r(v) \leq 0\} \\
&= \{v : r(v) = 0\}
\end{aligned}$$

Es ist $G(66000) = 5$ da $\underbrace{2^{2^{2^{2}}}}_{4} = 65536 \leq 66000 \leq 131072 = \underbrace{2^{2^{2^{2^{2}}}}}_{5}$

Eigenschaften der Ranggruppen

Eigenschaften der Ranggruppen

$$\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$$

Eigenschaften der Ranggruppen

$$\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$$
$$= \{v : \lfloor \log^{(j+1)} n \rfloor < r(v) \leq \lfloor \log^{(j)} n \rfloor\}$$

Eigenschaften der Ranggruppen

$$\begin{aligned}\gamma_j &:= \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}\\ &= \{v : \lfloor \log^{(j+1)} n \rfloor < r(v) \leq \lfloor \log^{(j)} n \rfloor\}\\ &= \{v : \lceil \log^{(j+1)} n \rceil \leq r(v) \leq \lfloor \log^{(j)} n \rfloor\}\end{aligned}$$

Abschätzung der Größen der Ranggruppen

Abschätzung der Größen der Ranggruppen

$|\gamma_j|$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i$$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i \leq \sum_{i=\log^{(j+1)} n}^{\log^{(j)} n} \frac{n}{2^i}$$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i \leq \sum_{i=\log^{(j+1)} n}^{\log^{(j)} n} \frac{n}{2^i}$$

$$\leq \sum_{i=\log^{(j+1)} n - \log^{(j+1)} n}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^{i+\log^{(j+1)} n}}$$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i \leq \sum_{i=\log^{(j+1)} n}^{\log^{(j)} n} \frac{n}{2^i}$$

$$\leq \sum_{i=\log^{(j+1)} n - \log^{(j+1)} n}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^{i+\log^{(j+1)} n}} = \sum_{i=0}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^i} \cdot \frac{1}{2^{\log^{(j+1)} n}}$$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i \leq \sum_{i=\log^{(j+1)} n}^{\log^{(j)} n} \frac{n}{2^i}$$

$$\leq \sum_{i=\log^{(j+1)} n - \log^{(j+1)} n}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^{i+\log^{(j+1)} n}} = \sum_{i=0}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^i} \cdot \frac{1}{2^{\log^{(j+1)} n}}$$

$$\leq \frac{n}{2^{\log^{(j+1)} n}} \Bigg(\underbrace{\sum_{i=0}^{\infty} \frac{1}{2^i}}_{\leq 2} \Bigg)$$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i \leq \sum_{i=\log^{(j+1)} n}^{\log^{(j)} n} \frac{n}{2^i}$$

$$\leq \sum_{i=\log^{(j+1)} n - \log^{(j+1)} n}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^{i+\log^{(j+1)} n}} = \sum_{i=0}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^i} \cdot \frac{1}{2^{\log^{(j+1)} n}}$$

$$\leq \frac{n}{2^{\log^{(j+1)} n}} \Big(\underbrace{\sum_{i=0}^{\infty} \frac{1}{2^i}}_{\leq 2} \Big)$$

$$\leq \frac{2n}{2^{\log^{(j+1)} n}} = \frac{2n}{2^{\log(\log^{(j)} n)}}$$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i \leq \sum_{i=\log^{(j+1)} n}^{\log^{(j)} n} \frac{n}{2^i}$$

$$\leq \sum_{i=\log^{(j+1)} n - \log^{(j+1)} n}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^{i+\log^{(j+1)} n}} = \sum_{i=0}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^i} \cdot \frac{1}{2^{\log^{(j+1)} n}}$$

$$\leq \frac{n}{2^{\log^{(j+1)} n}} \Big(\underbrace{\sum_{i=0}^{\infty} \frac{1}{2^i}}_{\leq 2} \Big)$$

$$\leq \frac{2n}{2^{\log^{(j+1)} n}} = \frac{2n}{2^{\log(\log^{(j)} n)}} = \frac{2n}{\log^{(j)} n \cdot 2^{\log 1}}$$

Abschätzung der Größen der Ranggruppen

$$|\gamma_j| = \sum_{\text{Raenge i in } \gamma_j} R_i \leq \sum_{i=\log^{(j+1)} n}^{\log^{(j)} n} \frac{n}{2^i}$$

$$\leq \sum_{i=\log^{(j+1)} n - \log^{(j+1)} n}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^{i+\log^{(j+1)} n}} = \sum_{i=0}^{\log^{(j)} n - \log^{(j+1)} n} \frac{n}{2^i} \cdot \frac{1}{2^{\log^{(j+1)} n}}$$

$$\leq \frac{n}{2^{\log^{(j+1)} n}} \Big(\underbrace{\sum_{i=0}^{\infty} \frac{1}{2^i}}_{\leq 2} \Big)$$

$$\leq \frac{2n}{2^{\log^{(j+1)} n}} = \frac{2n}{2^{\log(\log^{(j)} n)}} = \frac{2n}{\log^{(j)} n \cdot 2^{\log 1}} = \frac{2n}{\log^{(j)} n}$$

Abschätzung der Laufzeit von Union – Find

Es werden insgesamt m Operationen ausgeführt.

Abschätzung der Laufzeit von Union – Find

Es werden insgesamt m Operationen ausgeführt.

- Makeset hat Aufwand $O(1)$
  $\Rightarrow O(m)$ (Angenommen n mal $\Rightarrow n$ Knoten)

Abschätzung der Laufzeit von $\mathrm{Union} - \mathrm{Find}$

Es werden insgesamt m Operationen ausgeführt.

- $\mathrm{Makeset}$ hat Aufwand $O(1)$
  $\Rightarrow O(m)$ (Angenommen n mal $\Rightarrow n$ Knoten)
- Union hat Aufwand $O(1) \Rightarrow O(n)$

Abschätzung der Laufzeit von Union – Find

Es werden insgesamt m Operationen ausgeführt.

- Makeset hat Aufwand $O(1)$
  $\Rightarrow O(m)$ (Angenommen n mal $\Rightarrow n$ Knoten)
- Union hat Aufwand $O(1) \Rightarrow O(n)$
- Find $= \{$ Viele Neuzuweisungen von $p[v], p[p[v]], \ldots\}$

Abschätzung der Laufzeit von Union – Find

Es werden insgesamt m Operationen ausgeführt.

- Makeset hat Aufwand $O(1)$
  $\Rightarrow O(m)$ (Angenommen n mal $\Rightarrow n$ Knoten)
- Union hat Aufwand $O(1) \Rightarrow O(n)$
- Find = { Viele Neuzuweisungen von $p[v], p[p[v]], \ldots$}

Bezeichnung:

- Typ-A-Neuzuweisung von v: $p[v]$ in <u>anderer</u> Ranggruppe als v

Abschätzung der Laufzeit von Union – Find

Es werden insgesamt m Operationen ausgeführt.

- Makeset hat Aufwand $O(1)$
  $\Rightarrow O(m)$ (Angenommen n mal $\Rightarrow n$ Knoten)
- Union hat Aufwand $O(1) \Rightarrow O(n)$
- Find $=$ { Viele Neuzuweisungen von $p[v], p[p[v]], \ldots$}

Bezeichnung:

- Typ-A-Neuzuweisung von v: $p[v]$ in <u>anderer</u> Ranggruppe als v
- Typ-B-Neuzuweisung von v: $p[v]$ in <u>gleicher</u> Ranggruppe wie v

Die Typ-A Kosten

Die Typ-A Kosten

- Ränge streng monoton: $\mathrm{rank}(p[v]) > \mathrm{rank}(v)$

Die Typ-A Kosten

- Ränge streng monoton: $\mathrm{rank}(p[v]) > \mathrm{rank}(v)$
- $\Rightarrow$ Ranggruppen monoton: $p[v] \in \gamma_x, x \geq y, \mathrm{rank}(v) \in \gamma_y$

Die Typ-A Kosten

- Ränge streng monoton: $\mathrm{rank}(p[v]) > \mathrm{rank}(v)$
- $\Rightarrow$ Ranggruppen monoton: $p[v] \in \gamma_x, x \geq y, \mathrm{rank}(v) \in \gamma_y$
- Und: $|\text{Ranggruppen}| = G(n)$

Die Typ-A Kosten

- Ränge streng monoton: $\mathrm{rank}(p[v]) > \mathrm{rank}(v)$
- $\Rightarrow$ Ranggruppen monoton: $p[v] \in \gamma_x, x \geq y, \mathrm{rank}(v) \in \gamma_y$
- Und: $|\text{Ranggruppen}| = G(n)$

$\Longrightarrow$ Typ-A-Neuzuweisungen pro $\mathrm{Find} \leq G(n)$

Die Typ-A Kosten

- Ränge streng monoton: $\mathrm{rank}(p[v]) > \mathrm{rank}(v)$
- $\Rightarrow$ Ranggruppen monoton: $p[v] \in \gamma_x, x \geq y, \mathrm{rank}(v) \in \gamma_y$
- Und: $|\text{Ranggruppen}| = G(n)$

$\Longrightarrow$ Typ-A-Neuzuweisungen pro $\mathrm{Find} \leq G(n)$
$\Longrightarrow$ Typ-A-Neuzuweisungen insgesamt $\leq m \cdot G(n)$

Die Typ-A Kosten

- Ränge streng monoton: $\mathrm{rank}(p[v]) > \mathrm{rank}(v)$
- $\Rightarrow$ Ranggruppen monoton: $p[v] \in \gamma_x, x \geq y, \mathrm{rank}(v) \in \gamma_y$
- Und: $|\text{Ranggruppen}| = G(n)$

$\Longrightarrow$ Typ-A-Neuzuweisungen pro $\mathrm{Find} \leq G(n)$
$\Longrightarrow$ Typ-A-Neuzuweisungen insgesamt $\leq m \cdot G(n)$

$\Longrightarrow$ Typ-A-Kosten $\in O(m \cdot G(n))$

Die Typ-B Kosten

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$
- $\Rightarrow \exists \quad \log^{(j)} n - \log^{(j+1)} n \leq \log^{(j)} n$ verschiedene Ränge in γ_j

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$
- $\Rightarrow \exists \quad \log^{(j)} n - \log^{(j+1)} n \leq \log^{(j)} n$ verschiedene Ränge in γ_j
- $v \in \gamma_j$ kann maximal $\lfloor \log^{(j)n} \rfloor$ Typ-B-Neuzuw. erfahren.

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$
- $\Rightarrow \exists \quad \log^{(j)} n - \log^{(j+1)} n \leq \log^{(j)} n$ verschiedene Ränge in γ_j
- $v \in \gamma_j$ kann maximal $\lfloor \log^{(j)n} \rfloor$ Typ-B-Neuzuw. erfahren.
- Nach jeder Neuzuweisung: $\mathrm{rank}(p_{\mathrm{neu}}[v]) > \mathrm{rank}(p_{\mathrm{alt}}[v])$

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$
- $\Rightarrow \exists \quad \log^{(j)} n - \log^{(j+1)} n \leq \log^{(j)} n$ verschiedene Ränge in γ_j
- $v \in \gamma_j$ kann maximal $\lfloor \log^{(j)n} \rfloor$ Typ-B-Neuzuw. erfahren.
- Nach jeder Neuzuweisung: $\mathrm{rank}(p_{\mathrm{neu}}[v]) > \mathrm{rank}(p_{\mathrm{alt}}[v])$
- $\Rightarrow$ nach max. $\log^{(j)} n$ Typ-B eine Typ-A-Neuzuweisung

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$
- $\Rightarrow \exists \quad \log^{(j)} n - \log^{(j+1)} n \leq \log^{(j)} n$ verschiedene Ränge in γ_j
- $v \in \gamma_j$ kann maximal $\lfloor \log^{(j)n} \rfloor$ Typ-B-Neuzuw. erfahren.
- Nach jeder Neuzuweisung: $\mathrm{rank}(p_{\mathrm{neu}}[v]) > \mathrm{rank}(p_{\mathrm{alt}}[v])$
- $\Rightarrow$ nach max. $\log^{(j)} n$ Typ-B eine Typ-A-Neuzuweisung
- Ranggruppen monoton $\Rightarrow$ keine Typ-B-Neuzuw. mehr für v

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$
- $\Rightarrow \exists \quad \log^{(j)} n - \log^{(j+1)} n \leq \log^{(j)} n$ verschiedene Ränge in γ_j
- $v \in \gamma_j$ kann maximal $\lfloor \log^{(j)n} \rfloor$ Typ-B-Neuzuw. erfahren.
- Nach jeder Neuzuweisung: $\mathrm{rank}(p_{\mathrm{neu}}[v]) > \mathrm{rank}(p_{\mathrm{alt}}[v])$
- $\Rightarrow$ nach max. $\log^{(j)} n$ Typ-B eine Typ-A-Neuzuweisung
- Ranggruppen monoton $\Rightarrow$ keine Typ-B-Neuzuw. mehr für v
- $\forall v \in \gamma_j : \#\text{Typ-B}(v) \leq \log^{(j)} n$

Die Typ-B Kosten

- $\gamma_j := \{v : \log^{(j+1)} n < r(v) \leq \log^{(j)} n\}$
- $\Rightarrow \exists \quad \log^{(j)} n - \log^{(j+1)} n \leq \log^{(j)} n$ verschiedene Ränge in γ_j
- $v \in \gamma_j$ kann maximal $\lfloor \log^{(j)n} \rfloor$ Typ-B-Neuzuw. erfahren.
- Nach jeder Neuzuweisung: $\mathrm{rank}(p_{\mathrm{neu}}[v]) > \mathrm{rank}(p_{\mathrm{alt}}[v])$
- $\Rightarrow$ nach max. $\log^{(j)} n$ Typ-B eine Typ-A-Neuzuweisung
- Ranggruppen monoton $\Rightarrow$ keine Typ-B-Neuzuw. mehr für v
- $\forall v \in \gamma_j : \#\text{Typ-B}(v) \leq \log^{(j)} n$

summieren ...

Summe aller Typ-B Kosten

Summe aller Typ-B Kosten

#Typ-B-Neuzuweisungen für alle Knoten:

Summe aller Typ-B Kosten

#Typ-B-Neuzuweisungen für <u>alle</u> Knoten:

$$= \sum_{\text{Ranggruppen } \gamma_j} |\text{Knoten in } \gamma_j| \cdot (\text{Typ} - \text{B pro Knoten in } \gamma_j)$$

Summe aller Typ-B Kosten

#Typ-B-Neuzuweisungen für <u>alle</u> Knoten:

$$= \sum_{\text{Ranggruppen } \gamma_j} |\text{Knoten in } \gamma_j| \cdot (\text{Typ} - \text{B pro Knoten in } \gamma_j)$$

$$= \sum_{j=1}^{G(n)} |\gamma_j| \cdot \log^{(j)} n$$

Summe aller Typ-B Kosten

#Typ-B-Neuzuweisungen für <u>alle</u> Knoten:

$$= \sum_{\text{Ranggruppen } \gamma_j} |\text{Knoten in } \gamma_j| \cdot (\text{Typ} - \text{B pro Knoten in } \gamma_j)$$

$$= \sum_{j=1}^{G(n)} |\gamma_j| \cdot \log^{(j)} n$$

$$\leq \sum_{j=1}^{G(n)} \frac{2n}{\log^{(j)} n} \cdot \log^{(j)} n$$

Summe aller Typ-B Kosten

#Typ-B-Neuzuweisungen für <u>alle</u> Knoten:

$$= \sum_{\text{Ranggruppen } \gamma_j} |\text{Knoten in } \gamma_j| \cdot (\text{Typ} - \text{B pro Knoten in } \gamma_j)$$

$$= \sum_{j=1}^{G(n)} |\gamma_j| \cdot \log^{(j)} n$$

$$\leq \sum_{j=1}^{G(n)} \frac{2n}{\log^{(j)} n} \cdot \log^{(j)} n = \sum_{j=1}^{G(n)} 2n$$

Summe aller Typ-B Kosten

#Typ-B-Neuzuweisungen für alle Knoten:

$$= \sum_{\text{Ranggruppen } \gamma_j} |\text{Knoten in } \gamma_j| \cdot (\text{Typ} - \text{B pro Knoten in } \gamma_j)$$

$$= \sum_{j=1}^{G(n)} |\gamma_j| \cdot \log^{(j)} n$$

$$\leq \sum_{j=1}^{G(n)} \frac{2n}{\log^{(j)} n} \cdot \log^{(j)} n = \sum_{j=1}^{G(n)} 2n$$

$$= G(n) \cdot 2n$$

Summe aller Typ-B Kosten

#Typ-B-Neuzuweisungen für alle Knoten:

$$= \sum_{\text{Ranggruppen } \gamma_j} |\text{Knoten in } \gamma_j| \cdot (\text{Typ} - \text{B pro Knoten in } \gamma_j)$$

$$= \sum_{j=1}^{G(n)} |\gamma_j| \cdot \log^{(j)} n$$

$$\leq \sum_{j=1}^{G(n)} \frac{2n}{\log^{(j)} n} \cdot \log^{(j)} n = \sum_{j=1}^{G(n)} 2n$$

$$= G(n) \cdot 2n$$

Typ-B-Kosten $\in O(m \cdot G(n))$

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

- T_{Makeset}: $\in O(m)$ (Produziert n Sets, $n \leq m$)

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

- $T_{\text{Makeset}}: \in O(m)$ (Produziert n Sets, $n \leq m$)
- $T_{\text{Union}}: \in O(n)$

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

- T_{Makeset}: $\in O(m)$ (Produziert n Sets, $n \leq m$)
- T_{Union} : $\in O(n)$
- T_{Find}:

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

- $T_{\mathrm{Makeset}}: \in O(m)$ (Produziert n Sets, $n \leq m$)
- $T_{\mathrm{Union}}: \in O(n)$
- T_{Find}:
  - Typ-A-Kosten: $\in O(m \cdot G(n))$

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

- T_{Makeset}: $\in O(m)$ (Produziert n Sets, $n \leq m$)
- T_{Union} : $\in O(n)$
- T_{Find}:
  - Typ-A-Kosten: $\in O(m \cdot G(n))$
  - Typ-B-Kosten: $\in O(m \cdot G(n))$

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

- T_{Makeset}: $\in O(m)$ (Produziert n Sets, $n \leq m$)
- T_{Union} : $\in O(n)$
- T_{Find}:
  - Typ-A-Kosten: $\in O(m \cdot G(n))$
  - Typ-B-Kosten: $\in O(m \cdot G(n))$
- $T_{\text{Union}-\text{Find}} \in O(m \cdot G(n))$

Zusammenfassend:

Es werden insgesamt m Operationen ausgeführt.

- T_{Makeset}: $\in O(m)$ (Produziert n Sets, $n \leq m$)
- T_{Union} : $\in O(n)$
- T_{Find}:
  - Typ-A-Kosten: $\in O(m \cdot G(n))$
  - Typ-B-Kosten: $\in O(m \cdot G(n))$
- $T_{\text{Union-Find}} \in O(m \cdot G(n))$
- Da $n \leq m$ gilt auch $T_{\text{Union-Find}} \in O(m \cdot G(m))$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- T_{Makeset}: $\in O(m)$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- T_{Makeset}: $\in O(m)$
- T_{Union} : $\in O(m)$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- $T_{\mathrm{Makeset}}: \in O(m)$
- $T_{\mathrm{Union}} : \in O(m)$
- $T_{\mathrm{Find}}:$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- T_{Makeset}: $\in O(m)$
- T_{Union} : $\in O(m)$
- T_{Find}:
  - Beim ersten Find sind maximal m Knoten vorhanden.

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- $T_{\mathrm{Makeset}}: \in O(m)$
- $T_{\mathrm{Union}} : \in O(m)$
- T_{Find}:
  - Beim ersten Find sind maximal m Knoten vorhanden.
  - Find: jeder beteiligte Knoten v bekommt Preis 1

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- $T_{\mathrm{Makeset}}: \in O(m)$
- $T_{\mathrm{Union}}: \in O(m)$
- T_{Find}:
  - Beim ersten Find sind maximal m Knoten vorhanden.
  - Find: jeder beteiligte Knoten v bekommt Preis 1
  - Danach: Knoten v hat Höhe 1 $\Rightarrow$ $\mathrm{Find}(v)$ in $O(1)$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- $T_{\text{Makeset}}: \in O(m)$
- $T_{\text{Union}} : \in O(m)$
- T_{Find}:
  - Beim ersten Find sind maximal m Knoten vorhanden.
  - Find: jeder beteiligte Knoten v bekommt Preis 1
  - Danach: Knoten v hat Höhe 1 $\Rightarrow \text{Find}(v)$ in $O(1)$
  - Buchung von $\text{Find}(v)$ mit Tiefe$(v) = 1$ auf Konto von $\text{Find}(v)$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- $T_{\text{Makeset}}: \in O(m)$
- $T_{\text{Union}} : \in O(m)$
- T_{Find}:
  - Beim ersten Find sind maximal m Knoten vorhanden.
  - Find: jeder beteiligte Knoten v bekommt Preis 1
  - Danach: Knoten v hat Höhe 1 $\Rightarrow \text{Find}(v)$ in $O(1)$
  - Buchung von $\text{Find}(v)$ mit $\text{Tiefe}(v) = 1$ auf Konto von $\text{Find}(v)$
  - $\Rightarrow T_{\text{Find}}$ in $O(|\text{Knoten}| + |\text{Anzahl Find}|)$

Teilaufgabe a

Zu zeigen: Gesamtaufwand in $O(m)$

- $T_{\mathrm{Makeset}}: \in O(m)$
- $T_{\mathrm{Union}} : \in O(m)$
- T_{Find}:
  - Beim ersten Find sind maximal m Knoten vorhanden.
  - Find: jeder beteiligte Knoten v bekommt Preis 1
  - Danach: Knoten v hat Höhe $1 \Rightarrow \mathrm{Find}(v)$ in $O(1)$
  - Buchung von $\mathrm{Find}(v)$ mit $\mathrm{Tiefe}(v) = 1$ auf Konto von $\mathrm{Find}(v)$
  - $\Rightarrow T_{\mathrm{Find}}$ in $O(|\mathrm{Knoten}| + |\mathrm{Anzahl\ Find}|)$

$\Longrightarrow$ Gesamtaufwand in $O(m)$

Teilaufgabe b

Wie verhält es sich <u>ohne</u> Balancieren?

Teilaufgabe b

Wie verhält es sich ohne Balancieren?

Teilaufgabe a benutzt das Balancieren nicht!

Teilaufgabe b

Wie verhält es sich ohne Balancieren?

Teilaufgabe a benutzt das Balancieren nicht!
$\Rightarrow$ Immer noch: Gesamtaufwand in $O(m)$

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$.

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten
$\Rightarrow$ Gesamtaufwand in $O(m \log m)$

Teilaufgabe c

Wie verhält es sich <u>ohne</u> Pfadkompression aber mit Balancing?
$(T_{\mathrm{Makeset}}$ und $T_{\mathrm{Union}} : \in O(m))$

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten
$\Rightarrow$ Gesamtaufwand in $O(m \log m)$

Scharfes Beispiel:

Teilaufgabe c

Wie verhält es sich <u>ohne</u> Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten
$\Rightarrow$ Gesamtaufwand in $O(m \log m)$

Scharfes Beispiel:

- $m/3$ mal Makeset (m Knoten)

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten
$\Rightarrow$ Gesamtaufwand in $O(m \log m)$

Scharfes Beispiel:

- $m/3$ mal Makeset (m Knoten)
- $m/3$ mal Union (ein Baum mit Tiefe $\Theta(\log(m/3))$)

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten
$\Rightarrow$ Gesamtaufwand in $O(m \log m)$

Scharfes Beispiel:

- $m/3$ mal Makeset (m Knoten)
- $m/3$ mal Union (ein Baum mit Tiefe $\Theta(\log(m/3))$)
- $m/3$ mal Find auf tiefsten Knoten: $\Theta(\log(m/3))$ pro Find

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten
$\Rightarrow$ Gesamtaufwand in $O(m \log m)$

Scharfes Beispiel:

- $m/3$ mal Makeset (m Knoten)
- $m/3$ mal Union (ein Baum mit Tiefe $\Theta(\log(m/3))$)
- $m/3$ mal Find auf tiefsten Knoten: $\Theta(\log(m/3))$ pro Find

$\Rightarrow$ Gesamtaufwand in $\Omega((m/3) \log(m/3))$

Teilaufgabe c

Wie verhält es sich ohne Pfadkompression aber mit Balancing?
(T_{Makeset} und T_{Union} : $\in O(m)$)

Knotentiefe $\leq \log m$. Maximal m mal Find auf solchen Knoten
$\Rightarrow$ Gesamtaufwand in $O(m \log m)$

Scharfes Beispiel:

- $m/3$ mal Makeset (m Knoten)
- $m/3$ mal Union (ein Baum mit Tiefe $\Theta(\log(m/3))$)
- $m/3$ mal Find auf tiefsten Knoten: $\Theta(\log(m/3))$ pro Find

$\Rightarrow$ Gesamtaufwand in $\Omega((m/3)\log(m/3))$

$\Rightarrow$ Gesamtaufwand in $\Theta((m)\log(m))$

2-dimensional $\Longrightarrow$ 2 X 1-dimensional

Sei ohne Einschränkung $x_1 \leq x_2 \leq \cdots \leq x_n$
Gesucht ist ein Punkt $p_{\min} = (x, y)$ mit

$$p_{\min} = \min_p \sum_{i=1}^{n} w_i d(p, p_i)$$

2-dimensional $\Longrightarrow$ 2 X 1-dimensional

Sei ohne Einschränkung $x_1 \leq x_2 \leq \cdots \leq x_n$
Gesucht ist ein Punkt $p_{\min} = (x, y)$ mit

$$p_{\min} = \min_p \sum_{i=1}^{n} w_i d(p, p_i) = \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| + |y_p - y_i| \right) \right)$$

2-dimensional $\Longrightarrow$ 2 X 1-dimensional

Sei ohne Einschränkung $x_1 \leq x_2 \leq \cdots \leq x_n$
Gesucht ist ein Punkt $p_{\min} = (x, y)$ mit

$$
\begin{aligned}
p_{\min} = \min_p \sum_{i=1}^{n} w_i d(p, p_i) &= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| + |y_p - y_i| \right) \right) \\
&= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| \right) + \sum_{i=1}^{n} w_i \left(|y_p - y_i| \right) \right)
\end{aligned}
$$

2-dimensional $\Longrightarrow$ 2 X 1-dimensional

Sei ohne Einschränkung $x_1 \leq x_2 \leq \cdots \leq x_n$
Gesucht ist ein Punkt $p_{\min} = (x, y)$ mit

$$
\begin{aligned}
p_{\min} = \min_p \sum_{i=1}^{n} w_i d(p, p_i) &= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| + |y_p - y_i| \right) \right) \\
&= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| \right) + \sum_{i=1}^{n} w_i \left(|y_p - y_i| \right) \right) \\
&= \min_p \left(\sum_{i=1}^{n} w_i \Delta_x \right) + \min_p \left(\sum_{i=1}^{n} w_i \Delta_y \right)
\end{aligned}
$$

2-dimensional $\Longrightarrow$ 2 X 1-dimensional

Sei ohne Einschränkung $x_1 \leq x_2 \leq \cdots \leq x_n$
Gesucht ist ein Punkt $p_{\min} = (x, y)$ mit

$$\begin{aligned}
p_{\min} = \min_p \sum_{i=1}^{n} w_i d(p, p_i) &= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| + |y_p - y_i| \right) \right) \\
&= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| \right) + \sum_{i=1}^{n} w_i \left(|y_p - y_i| \right) \right) \\
&= \min_p \left(\sum_{i=1}^{n} w_i \Delta_x \right) + \min_p \left(\sum_{i=1}^{n} w_i \Delta_y \right)
\end{aligned}$$

Betrachten also zwei unabängige, 1-dimensionale Probleme.

2-dimensional $\Longrightarrow$ 2 X 1-dimensional

Sei ohne Einschränkung $x_1 \leq x_2 \leq \cdots \leq x_n$
Gesucht ist ein Punkt $p_{\min} = (x, y)$ mit

$$
\begin{aligned}
p_{\min} = \min_p \sum_{i=1}^{n} w_i d(p, p_i) &= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| + |y_p - y_i| \right) \right) \\
&= \min_p \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| \right) + \sum_{i=1}^{n} w_i \left(|y_p - y_i| \right) \right) \\
&= \min_p \left(\sum_{i=1}^{n} w_i \Delta_x \right) + \min_p \left(\sum_{i=1}^{n} w_i \Delta_y \right)
\end{aligned}
$$

Betrachten also zwei unabängige, 1-dimensionale Probleme.

Gesucht ist ein Punkt $x_{\min}$ mit $x_{\min} = \min_x \left(\sum_{i=1}^{n} w_i \left(|x_p - x_i| \right) \right)$

1-dimensional (1)

$W := 1/2 \sum_{i=1}^{n} w_i$

1-dimensional (1)

$W := 1/2 \sum_{i=1}^{n} w_i$

Für den gewichteten Median x_m der x-Werte gilt:

1-dimensional (1)

$W := 1/2 \sum_{i=1}^{n} w_i$

Für den gewichteten Median x_m der x-Werte gilt:

$$\sum_{x_i < x_m} w_i \leq W \geq \sum_{x_i > x_m} w_i$$

1-dimensional (1)

$W := 1/2 \sum_{i=1}^{n} w_i$

Für den gewichteten Median x_m der x-Werte gilt:

$$\sum_{x_i < x_m} w_i \leq W \geq \sum_{x_i > x_m} w_i$$

OBdA rechte Seite von x_m : $\sum_{x_i < x_m} w_i + w_m \geq W$

1-dimensional (1)

$W := 1/2 \sum_{i=1}^{n} w_i$

Für den gewichteten Median x_m der x-Werte gilt:

$$\sum_{x_i < x_m} w_i \leq W \geq \sum_{x_i > x_m} w_i$$

OBdA rechte Seite von x_m : $\sum_{x_i < x_m} w_i + w_m \geq W$

Laufe um kleines Δ nach rechts zu $\tilde{x} = x_m + \Delta$

1-dimensional (1)

$W := 1/2 \sum_{i=1}^{n} w_i$

Für den gewichteten Median x_m der x-Werte gilt:

$$\sum_{x_i < x_m} w_i \leq W \geq \sum_{x_i > x_m} w_i$$

OBdA rechte Seite von x_m : $\quad \sum_{x_i < x_m} w_i + w_m \geq W$

Laufe um kleines Δ nach rechts zu $\tilde{x} = x_m + \Delta$

$$\tilde{W} \geq W_m + \left(\sum_{x_i < x_m} w_i + w_m \right) \cdot \Delta - \left(\sum_{x_i > x_m} w_i \right) \cdot \Delta$$

1-dimensional (1)

$W := 1/2 \sum_{i=1}^{n} w_i$
Für den gewichteten Median x_m der x-Werte gilt:

$$\sum_{x_i < x_m} w_i \leq W \geq \sum_{x_i > x_m} w_i$$

OBdA rechte Seite von x_m : $\sum_{x_i < x_m} w_i + w_m \geq W$
Laufe um kleines Δ nach rechts zu $\tilde{x} = x_m + \Delta$

$$\tilde{W} \geq W_m + \left(\sum_{x_i < x_m} w_i + w_m \right) \cdot \Delta - \left(\sum_{x_i > x_m} w_i \right) \cdot \Delta$$

$$\geq W_m + \underbrace{\left(\sum_{x_i < x_m} w_i + w_m - \sum_{x_i > x_m} w_i \right)}_{\geq 0} \cdot \Delta$$

1-dimensional (2)

Ab dem nächsten Punkt x_{m+1} also ab $(\Delta - \Delta_{m,m+1}) \geq 0$ gilt ähnliches:

1-dimensional (2)

Ab dem nächsten Punkt x_{m+1} also ab $(\Delta - \Delta_{m,m+1}) \geq 0$ gilt ähnliches:

$$\tilde{W} = W_m + \underbrace{\left(\sum_{x_i < x_m} w_i + w_m - \sum_{x_i > x_{m+1}} w_i\right) \cdot \Delta - \underbrace{w_{m+1} \cdot \Delta_{m,m+1}}_{\text{über } \Delta_{m,m+1} \text{ war } x_{m+1} \text{ rechts}}}_{\geq 0 \text{ wie vorher (sogar schlimmer)}}$$

$$+ \underbrace{w_{m+1} \cdot \underbrace{(\Delta - \Delta_{m,m+1})}_{\geq 0}}_{\text{nun } x_{m+1} \text{ links}}$$

1-dimensional (2)

Ab dem nächsten Punkt x_{m+1} also ab $(\Delta - \Delta_{m,m+1}) \geq 0$ gilt ähnliches:

$$\tilde{W} = W_m + \underbrace{\left(\sum_{x_i < x_m} w_i + w_m - \sum_{x_i > x_{m+1}} w_i\right) \cdot \Delta - \underbrace{w_{m+1} \cdot \Delta_{m,m+1}}_{\text{über } \Delta_{m,m+1} \text{ war } x_{m+1} \text{ rechts}}}_{\geq 0 \text{ wie vorher (sogar schlimmer)}}$$

$$+ \underbrace{w_{m+1} \cdot \underbrace{(\Delta - \Delta_{m,m+1})}_{\geq 0}}_{\text{nun } x_{m+1} \text{ links}}$$

$$\geq W_m$$

1-dimensional (2)

Ab dem nächsten Punkt x_{m+1} also ab $(\Delta - \Delta_{m,m+1}) \geq 0$ gilt ähnliches:

$$\tilde{W} = W_m + \underbrace{\left(\sum_{x_i < x_m} w_i + w_m - \sum_{x_i > x_{m+1}} w_i\right) \cdot \Delta - \underbrace{w_{m+1} \cdot \Delta_{m,m+1}}_{\text{über } \Delta_{m,m+1} \text{ war } x_{m+1} \text{ rechts}}}_{\geq 0 \text{ wie vorher (sogar schlimmer)}}$$

$$+ \underbrace{w_{m+1} \cdot \underbrace{(\Delta - \Delta_{m,m+1})}_{\geq 0}}_{\text{nun } x_{m+1} \text{ links}}$$

$$\geq W_m$$

Alle Punkte, welche gewichtete Mediane sind, sind Lösungen

1-dimensional (2)

Ab dem nächsten Punkt x_{m+1} also ab $(\Delta - \Delta_{m,m+1}) \geq 0$ gilt ähnliches:

$$\tilde{W} = W_m + \underbrace{\left(\sum_{x_i < x_m} w_i + w_m - \sum_{x_i > x_{m+1}} w_i \right) \cdot \Delta - \underbrace{w_{m+1} \cdot \Delta_{m,m+1}}_{\text{über } \Delta_{m,m+1} \text{ war } x_{m+1} \text{ rechts}}}_{\geq 0 \text{ wie vorher (sogar schlimmer)}}$$

$$+ \underbrace{w_{m+1} \cdot \underbrace{(\Delta - \Delta_{m,m+1})}_{\geq 0}}_{\text{nun } x_{m+1} \text{ links}}$$

$$\geq W_m$$

Alle Punkte, welche gewichtete Mediane sind, sind Lösungen

$\Rightarrow$ Löse für x- und y-Achse mit $\mathrm{G-Median}$ in $O(n)$

1-dimensional (2)

Ab dem nächsten Punkt x_{m+1} also ab $(\Delta - \Delta_{m,m+1}) \geq 0$ gilt ähnliches:

$$\tilde{W} = W_m + \underbrace{\left(\sum_{x_i < x_m} w_i + w_m - \sum_{x_i > x_{m+1}} w_i\right) \cdot \Delta - \underbrace{w_{m+1} \cdot \Delta_{m,m+1}}_{\text{über } \Delta_{m,m+1} \text{ war } x_{m+1} \text{ rechts}}}_{\geq 0 \text{ wie vorher (sogar schlimmer)}}$$

$$+ \underbrace{w_{m+1} \cdot \underbrace{(\Delta - \Delta_{m,m+1})}_{\geq 0}}_{\text{nun } x_{m+1} \text{ links}}$$

$$\geq W_m$$

Alle Punkte, welche gewichtete Mediane sind, sind Lösungen

$\Rightarrow$ Löse für x- und y-Achse mit $\mathrm{G-Median}$ in $O(n)$

$\Rightarrow$ $p = (x_{\min}, y_{\min})$ ist optimale Postamtplatzierung

Äquivalenzklassen in Graustufenbildern

Anderes

- Kommentare zur Korrektur?
- Kommentare zur Übung?
- Kommentare zur Vorlesung?
- Übungsblätter machen!
- Musterlösung ab morgen online.
- Übungsblätter zurück am Dienstag.