Beweis:

1.) • Jeder Knoten $v \in C_1(s)$ ist Kind eines Knotens auf dem Suchpfad nach $x_1(s)$
—— " —— $v \in C_2(s)$ —— " —— $x_2(s)$

2.) • Das Gerüst des Baumes hat Platzbedarf $\Theta(N)$ (dh linearer Platz)
1.) ⇒ jedes Segment ist in $O(2 \log N)$ Knotenlisten gespeichert
⇒ Platzbedarf: $\Theta(N + n \cdot \log N)$ (n: # Segmente, dh n = maximale Größe der Knotenliste)

3/4) • 1.) ⇒ jedes Segment muss aus $O(2 \log N)$ Knotenlisten entfernt werden

- ## Suchen in Segmentbäumen:

Wir suchen $x \in \mathbb{R}$

Sei P der Suchpfad nach x
Dann gilt:
- $|P \cap C(s)| \leq 1 \quad \forall s \in S$
- $|P \cap C(s)| = 1 \quad x_1(s) \leq x \leq x_2(s)$
- $\{s \in S: x_1(s) \leq x \leq x_2(s)\} = \bigcup_{v \in P} NL(v)$

Dh die Knotenlisten auf dem Suchpfad nach x enthalten die Segmente die von einer vertikalen Geraden durch x geschnitten werden.
Die Menge $S(x) := \{s \in S: x_1(s) \leq x \leq s_2(x)\}$ kann in Zeit $O(\log N + m)$ berechnet werden,
wobei $m = |S(x)|$ dh die Größe der Ausgabe ist

- ## Suchproblem:

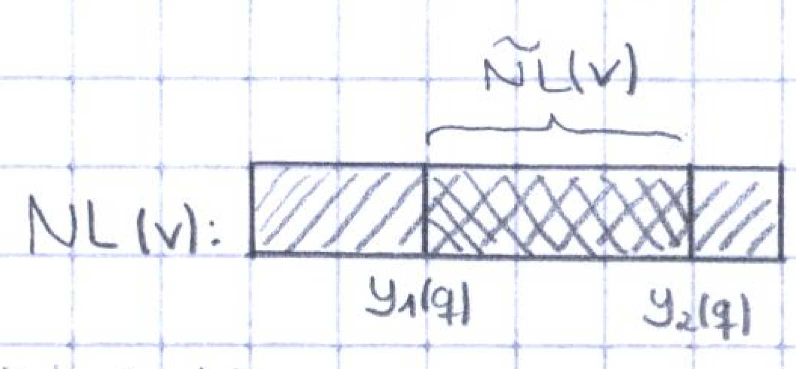
- Geg.: Vertikales Segment q mit:
  - $y_1(q)$: y-Koord des unteren Endpkts von q
  - $y_2(q)$: —— " —— oberen —— " ——
  - $x(q)$: x-Koord von q
- Ges.: $S(q) := \{s \in S: s \cap q \neq \emptyset\}$

- $S(q) = \{s \in S: x_1(s) \leq x(q) \leq x_2(s) \text{ und } y_1(q) \leq y(s) \leq y_2(q)\} =$
$= \bigcup_{v \in P} \tilde{NL}(v)$

mit $\tilde{NL}(v) := \{s \in NL(v): y_1(q) \leq y(s) \leq y_2(q)\}$

- Algorithmus zur Berechnung von $S(q)$:

P ← Suchpfad nach $x(q)$
Forall $v \in P$ do
    Lokalisiere $y_1(q)$ und $y_2(q)$ in $NL(v)$
    Gib alle Segmente dazwischen aus
od



Berechnung von $S(q)$ kostet: $O(\log N \cdot \overbrace{h(n)}^{\text{Suchzeit in Knotenliste der Länge } n} + k)$ k: Größe der Ausgabe

- ## Realisierung der Knotenlisten durch binäre balancierte Bäume:

Für Knotenlisten der Länge n gilt:
- Platz: $\Theta(n)$
- Einfügen: $f(n) = \Theta(\log n)$
- Streichen: $g(n) = \Theta(\log n)$
- Suchen: $h(n) = \Theta(\log n)$

- ## Satz:

- Der Segmentbaum braucht Platz $\Theta(N + n \cdot \log N)$
- Einfügen / Streichen eines Segments kostet: $\Theta(\log N \cdot \log n)$
- Suchen nach allen von einem vertikalen Segment geschnittenen Segmenten kostet: $\Theta(\log n \cdot \log N + m)$

Suchen in Knotenliste   Pfad   Ausgabe