Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Ball Partitioning / Generalized Hyperplanepartititoning

Inhalt

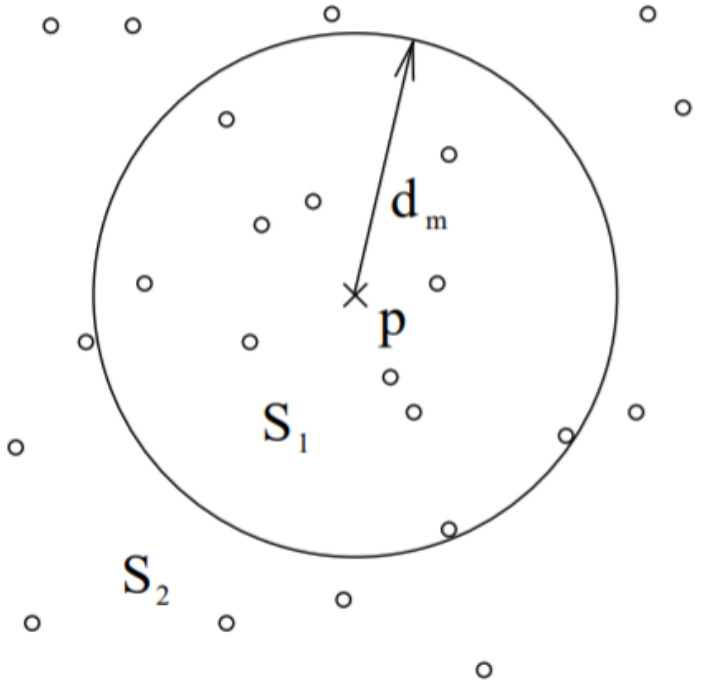
1. Ball Partitioning / Generalized Hyperplanepartitioning
2. K-d-Tree
3. VP-tree (BS-tree)
    1. nearest-neighbor
    2. splitting
    3. merging
4. Quellen

Ball Partitioning / Generalized Hyperplanepartitioning

- Problemstellung: Schwierigkeiten in der Indexierung, da Partitioninerung auf absoluten Koordinaten des Vektorraums basiert.
- Lösung: Aufteilung in **dynamische** Räume
- Aufteilung des Subsets S in S_1 und S_2 durch Median und Pivot

Ball Partitioning

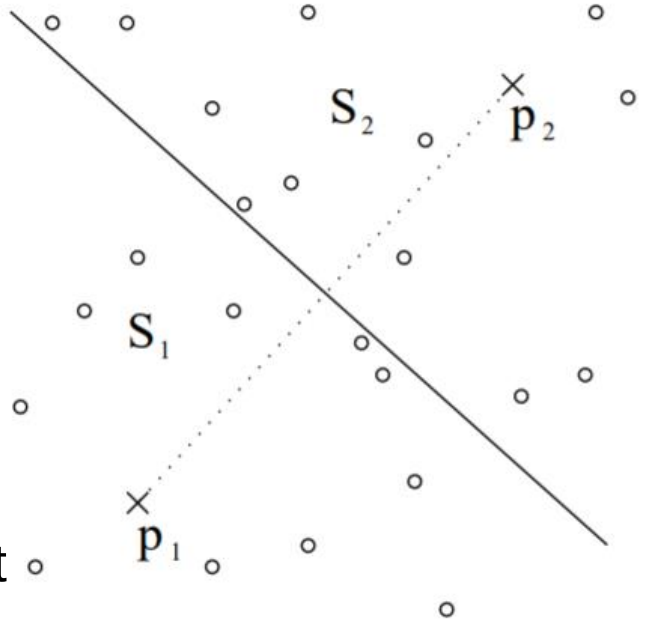
- Sphärischer Schnitt
- $p \in D$ p als Pivot, D als Daten
- d_m als Median der Menge $\{d(o_i, p), \forall o_i \in$
- Verteilung aller $o_j \in S$ in S_1 und S_2 nach:
  - $S_1 \leftarrow \{o_j \mid d(o_j, p) \leq d_m\}$
  - $S_2 \leftarrow \{o_j \mid d(o_j, p) \geq d_m\}$



[5]: S.29, Ball Partitioning

Generalized Hyperplanepartitioning

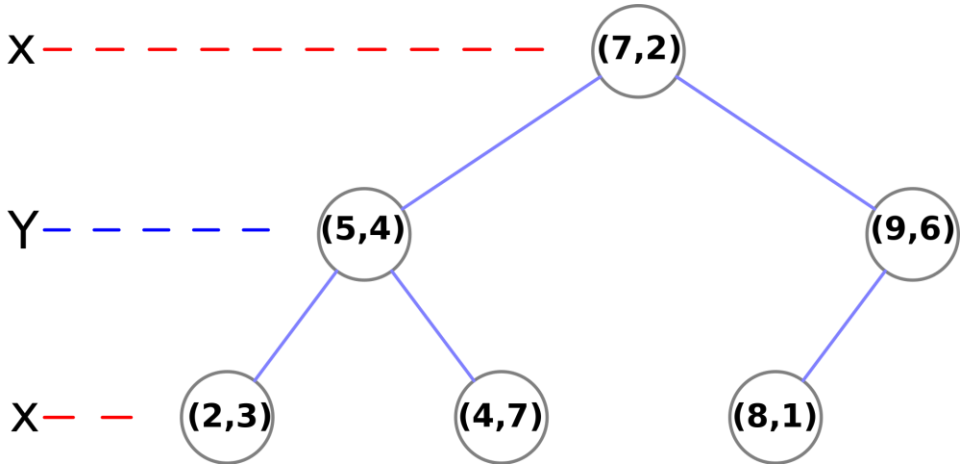
- Schnitt der Datenmenge
- $p_1, p_2 \in D$ p_1 und p_2 als Pivot, D als Daten
- Verteilung aller $o_j \in S$ in S_1 und S_2 nach:
  - $S_1 \leftarrow \{o_j \mid d(p_1, o_j) \leq d(p_2, o_j)\}$
  - $S_2 \leftarrow \{o_j \mid d(p_1, o_j) \geq d(p_2, o_j)\}$
- Weniger dynamisch, da kein Median bestimmt wird -> keine Balance gewährleistet

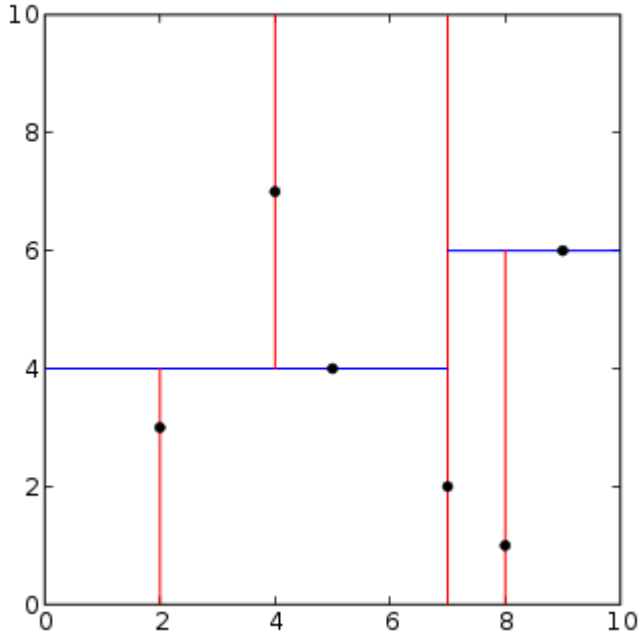


[5]: S.29, Generalized Hyperplanepartitioning

K-d-tree

- K = Anzahl
- d = Dimensionen





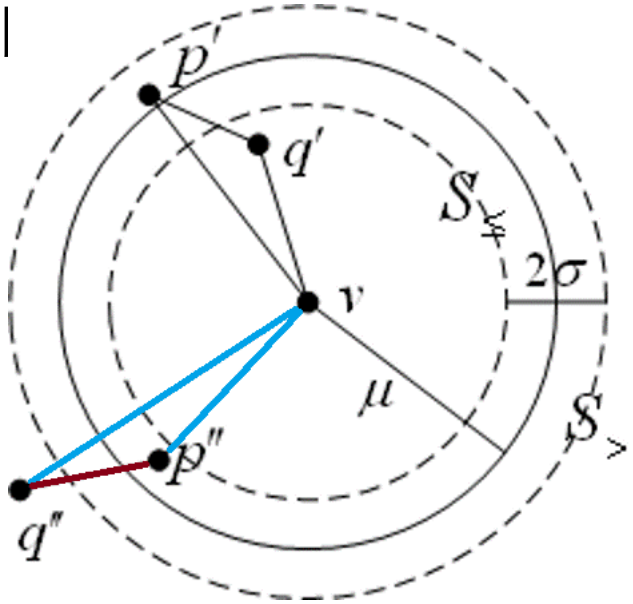
Quelle: https://en.wikipedia.org/wiki/K-d_tree

VP-tree (BS-tree)

- Aufteilen des Suchraums durch relative Distanzen zwischen vantage points und derer Kinder
- Berechnung der Entfernung zwischen Punkt und zugehörigem Raum einfach
- Feature Extraktion
  - Zusammenfassen zu Feature Klassen
  - Z.B. Farben, Textur, Form, …
- (Mind. so gut wie k-d-tree, effizienter als R*-tree und M-tree)
- Präziser aufgrund Raumaufteilung in Sphären statt in Vielecke
  - Durch Vergrößerung des Raums steigt das Volumen weniger stark in Relation

VP-tree (BS-tree) – nearest-neighl

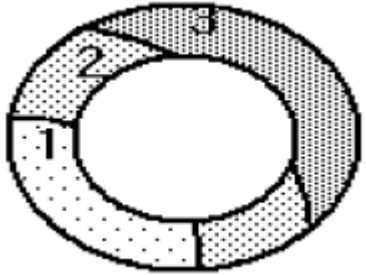
- Sphäre S_D als Teilraum von S
- Vantage point v
- $p \in S_D - \{v\}$ und Median für jeden anderen Punkt
- Subset $S_>$ für Punkte $p \in \{S_D\text{-}\mu > S_D\text{-}v\}$
- Subset $S_\leq$ für Punkte $p \in \{S_D\text{-}\mu \leq S_D\text{-}v\}$
- Punkte q‘, q‘‘
- σ als Schwelle
- Falls $d(v,q) \leq \mu - \sigma \rightarrow S_\leq$ $d(q,p) \geq |d(v,q) - d(v,p)| > |(\mu + \sigma) - \mu| = \sigma$
- Falls $d(v,q) > \mu + \sigma \rightarrow S_>$ $d(q,p) \geq |d(v,p) - d(v,q)| > |\mu - (\mu - \sigma)| = \sigma$



[4]: Seite 2, Data set partitioning and nearest-neighbor search

VP-tree (BS-tree)

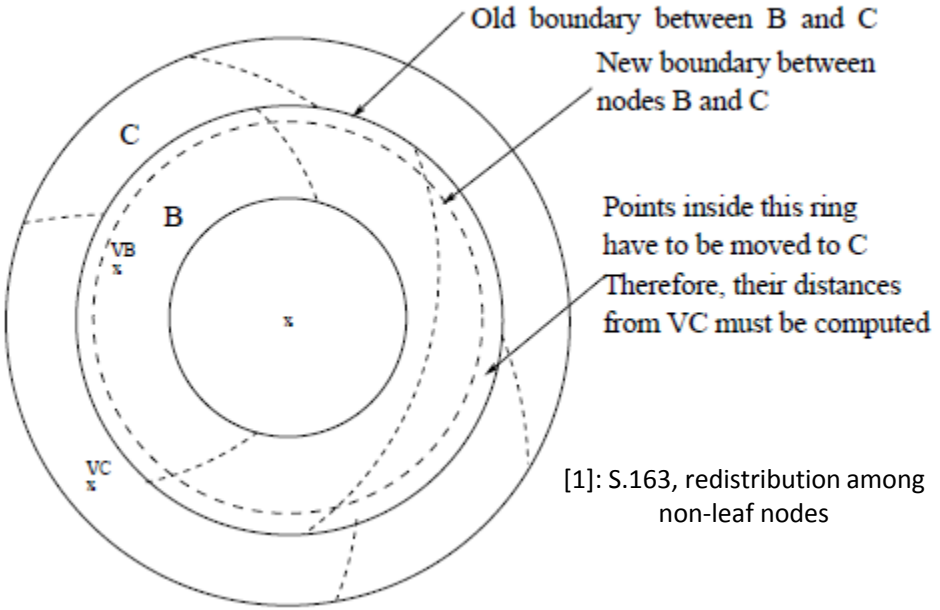
- Mehr als 2 Aufteilungen möglich
- „multivantage point tree“
- Problem: Rechenaufwand steigt je mehr Schichten dazukommen, da diese immer dünner werden!
- Suchoperatoren müssen in mehr als einer Schicht suchen



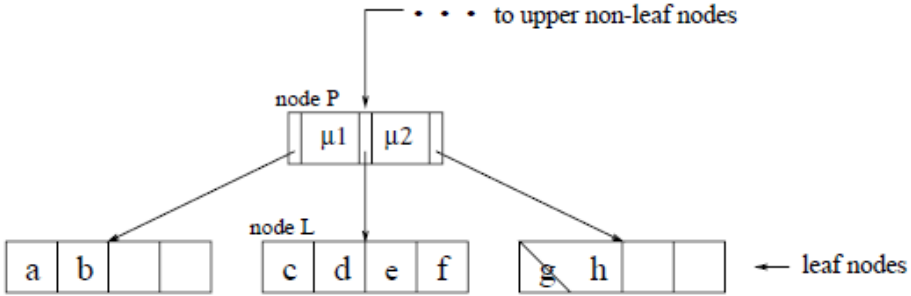
[2]: S.373, spherical shell partitioning using vp from outside

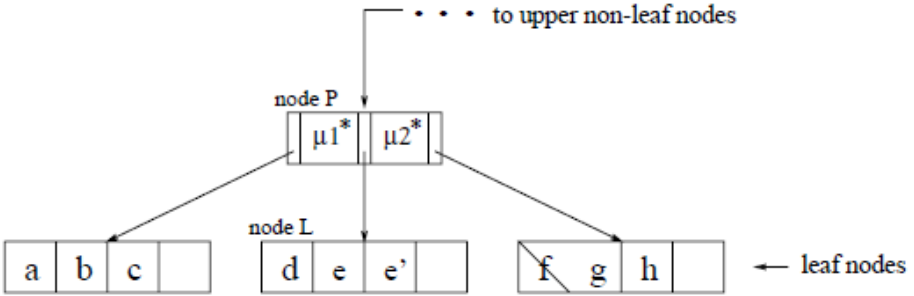
VP-tree (BS-tree) - splitting

- Elemente in B und C werden gleichmäßig aufgeteilt

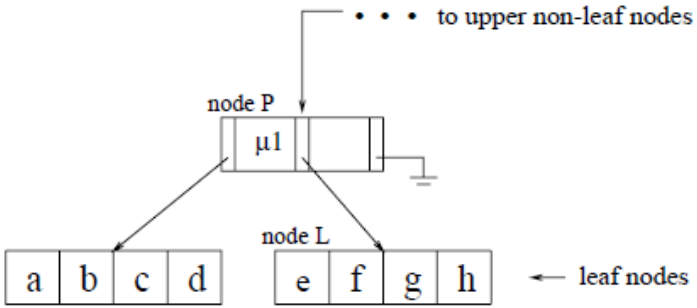


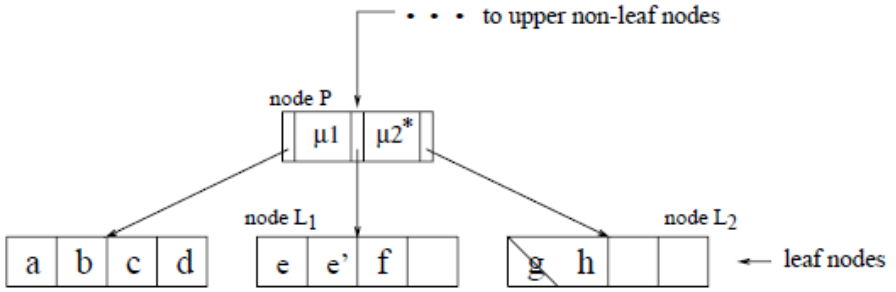
[1]: S.163, redistribution among non-leaf nodes



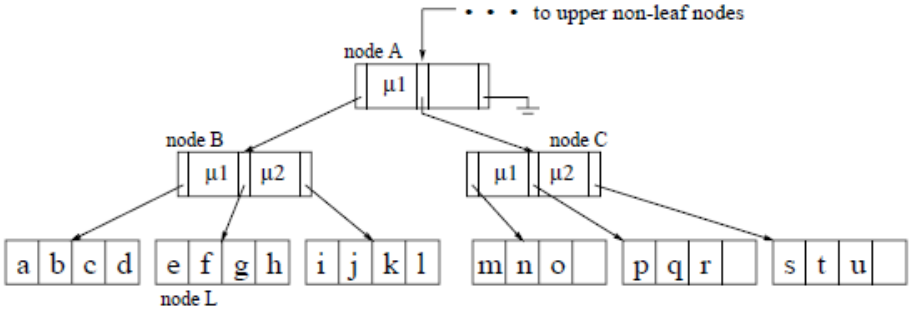


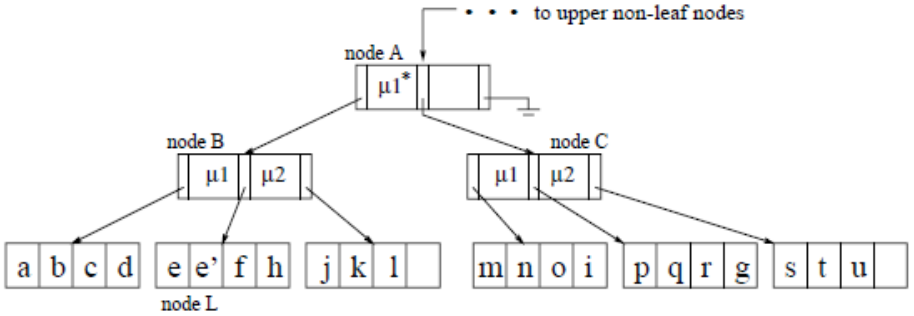
- e‘ soll eingefügt werden
- Neuverteilung unter den Kindern von P, da genug Platz
- Updaten der Grenzen



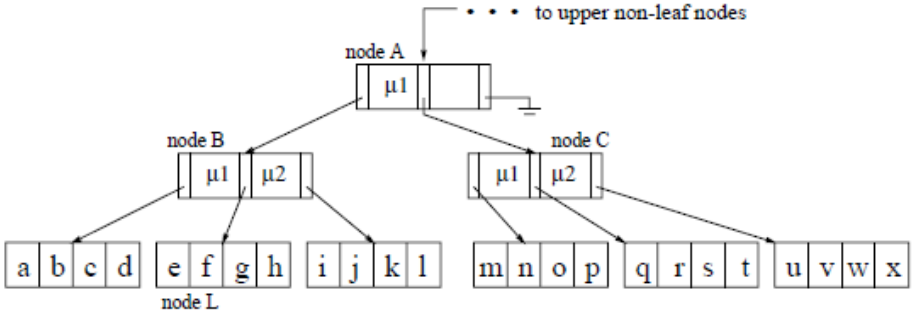


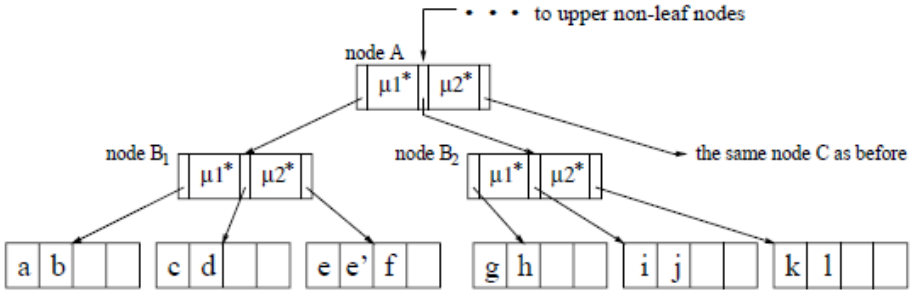
- e‘ soll eingefügt werden
- Neuverteilung unter den Kindern von P nicht möglich, da nicht genug Platz
- Erschaffen eines neuen Knotens
- Updaten der Grenzen



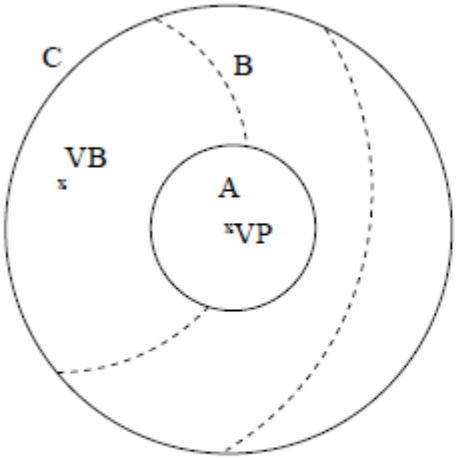


- e‘ soll in L eingefügt werden
- B ist voll
  - Neuverteilung unter B und C
- Gleichmäßige Verteilung: 12 zu 9 -> 11 zu 11
  - h und i am weitesten vom VP von A entfernt

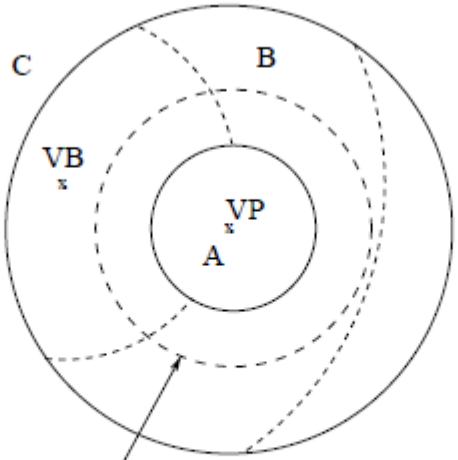




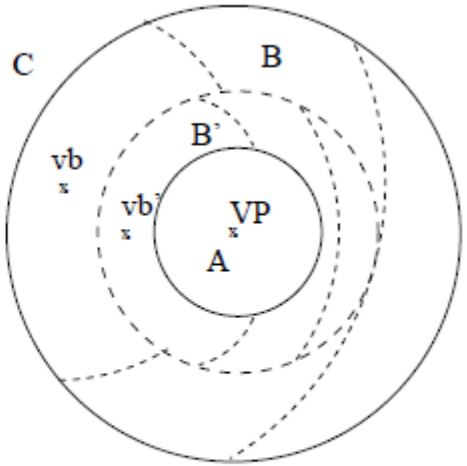
- e‘ soll in L eingefügt werden
- B und C sind voll
  - Split von B in B1 und B2



Node B has to be splitted



Splitting boundary (using VP as the vantage point).

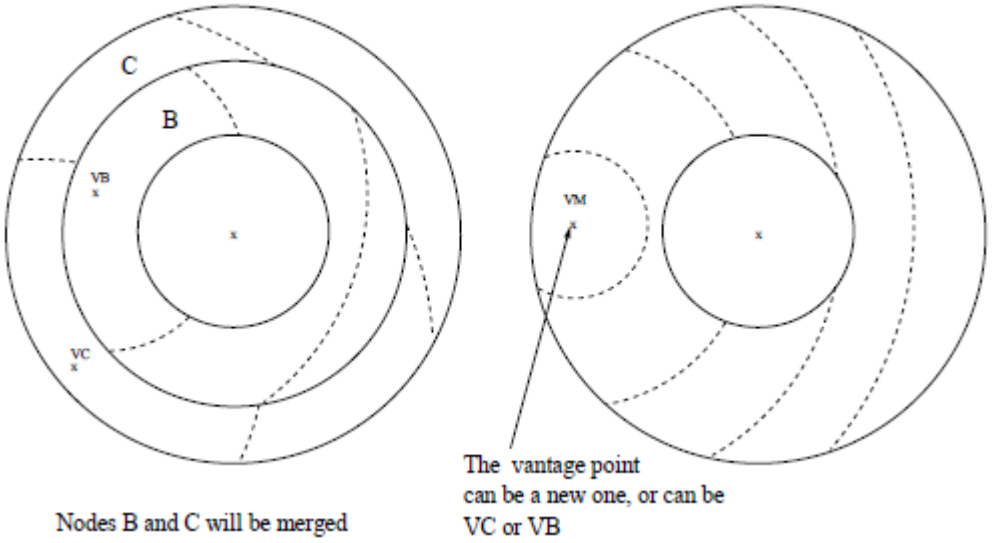


Result of the splitting new node B' created

[1]: S.165, Split of node B

VP-tree (BS-tree) - mergi

- Elemente in B und C können auch vereint werden
- B hat einen Elternknoten A und F-1 Geschwister (F=Gesamtmenge Knoten)
    1. Wenn Geschwisterraum von B alle Elemente von B aufnehmen kann
    2. Falls der Platz nicht für alle Elemente von ausreicht, werden die Elemente unter A neu verteilt -> F Knoten



[1]: S.166, merging two nodes

Quellen

[1]: **Dynamic vp-tree indexing for *n*-nearest neighbor search given pair-wise distances;** Ada Wai-chee Fu, Polly Mei-shuen Chan, Yin-Ling Cheung, Yiu Sang Moon

[2]: **Indexing Large Metric Spaces for Similarity Search Queries;** Tolga Bozkaya, Meral Ozsoyoglu

[3]: **Pivot-based Metric Indexing: Experiments and Analyses;** Lu Chen, Yunjun Gao, Baihua Zheng, Christian S. Jensen, Hanyu Yang, Keyu Yang

[4]: **VP-tree: Content-Based Image Indexing;** Il'ya Markov TP

[5]: **Scalable and Distributed Similarity Search;** Michal Batko