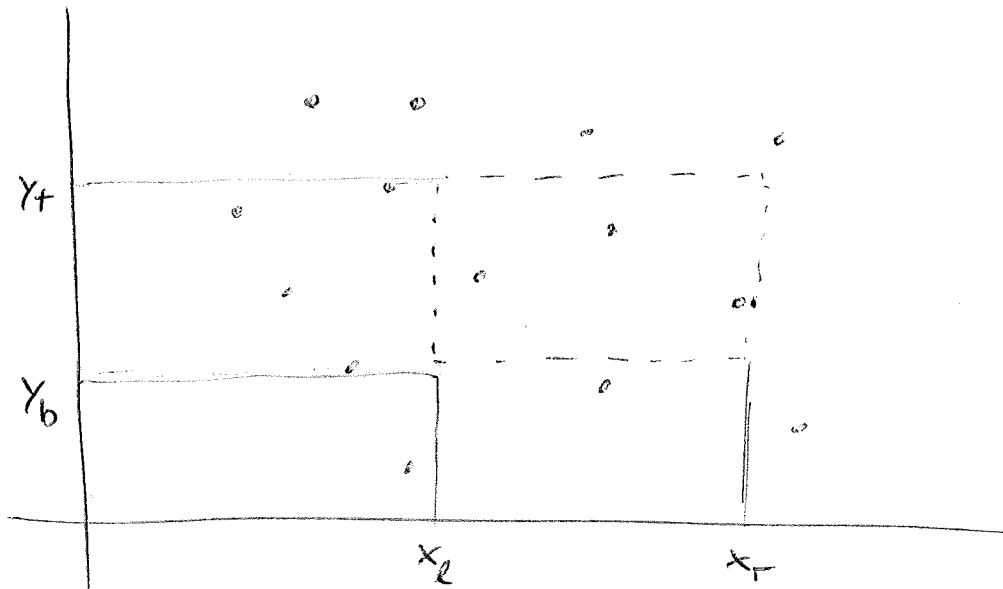


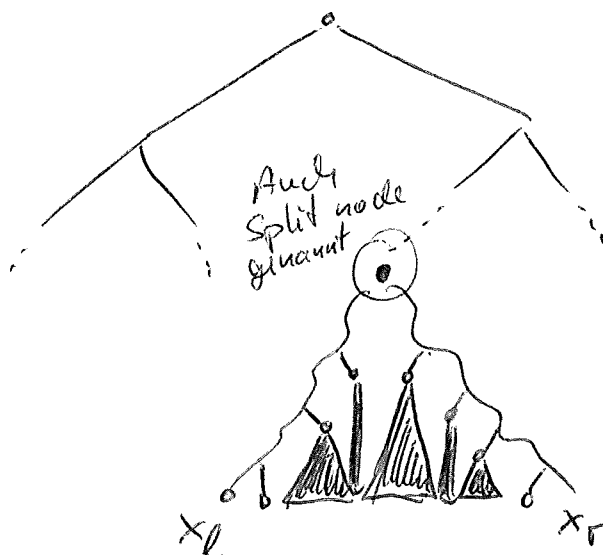
Beispiel kDtree + Quadtrees

- je 1 Studi zeichnet Punktmenge für kDtree/Quadtrees
- je 1 Studi zeichnet Tree
- je 1 Studi führt Range Query aus.

Range Trees



1D Query auf x :



Damit können wir zumindest, welche Punkte die x -Koordinate erfüllen.

Wir müssen nun jeden Teilbaum, der vollständig in der Range liegt durchsuchen, ob die y -Koordinate erfüllt ist.

Um dort Punkte schnell ausschließen zu können, benötigen wir eine weitere Range query. Aber nur auf Punkten des Teilbaums der ersten Query. (1.2)

Idee: Erstelle einen 1D-Baum ^{der y-Koord.} bzgl. der im Teilbaum ~~enthaltenen Punkte~~ des 1D-Baums ^{bzgl. x-Koord.} enthaltenen Punkte

→ jeder ~~Knoten~~ ^{Punkt} kommt in $O(\log n)$ solchen Bäumen vor.

→ Speicherbedarf wächst auf $O(n \log n)$.

Außerdem: Wir müssen nur $O(\log n)$ Range queries bzgl. der y-Koordinate durchführen

→ Range Query benötigt Zeit $O(\log^2(n) + k)$ Zeit.

Allgemein:

Bau ein 1D-Tree bzgl. 1. Koordinate und baue für jeden Knoten rekursiv einen RangTree ohne 1. Koordinate

→ Speicherbedarf $O(n \log^d n)$

→ Query Time $O(\log^d n + k)$.