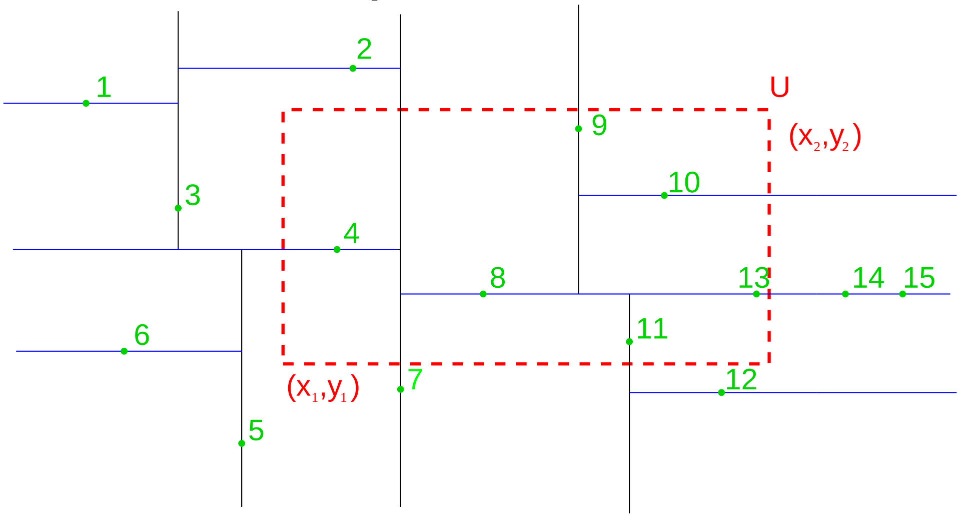
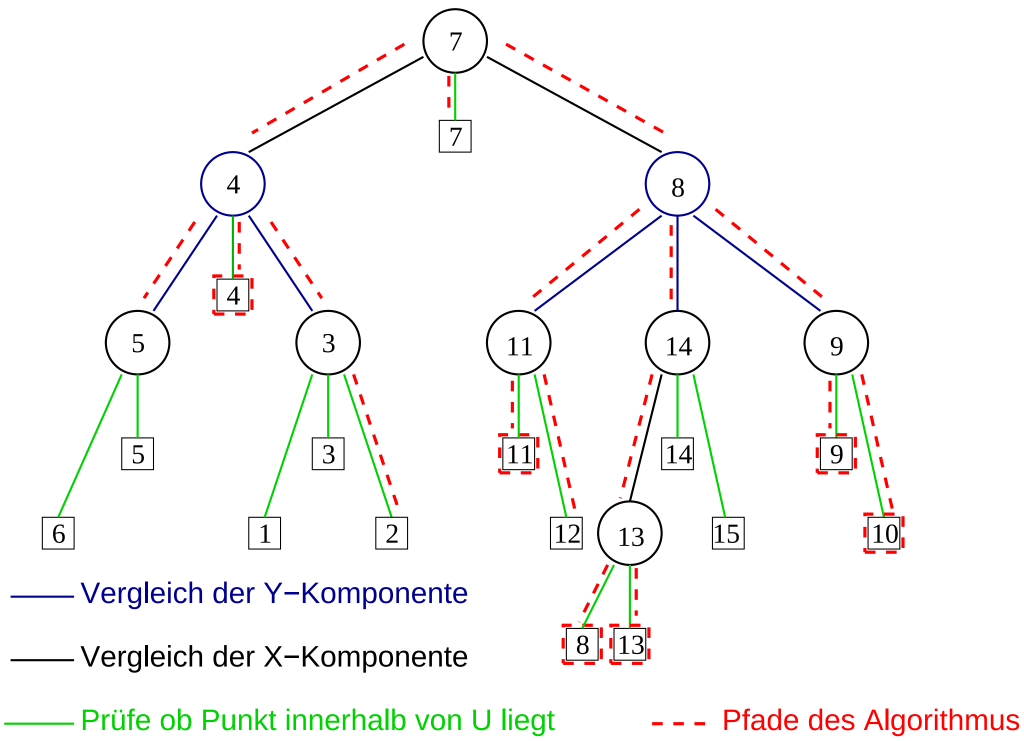
Beispiel zu 2-d-Bäumen



Es ergibt sich folgender 2-d-Baum



Laufzeit des Suchalgorithmus

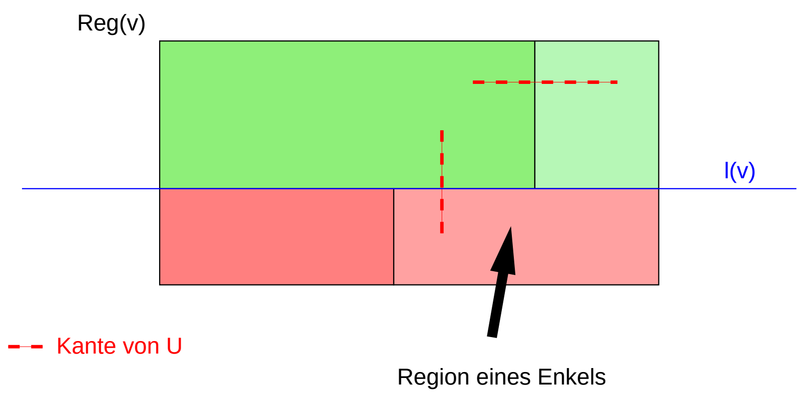
Zur Analyse betrachtet man die besuchten Knoten des Algorithmus

> a) Für jeden Knoten v, mit dessen Region (oder seiner Kinder Regionen) U einen nicht leeren Schnitt bildet, benötigt der Algorithmus konstante Zeit. Da die Anzahl der Knoten mit $Reg(v) \subset U = m$ ist, braucht man für diese Knoten insgesamt eine Laufzeit von $O(m)$.

Für die restlichen Knoten des Baumes betrachtet man eine beliebige aber feste Seite L von U.

Def: k ist **primärer Knoten** eines k-d-Baumes falls jede Kante auf dem Pfad zu k keine Gleichkante (=-Kante) ist.

> b) Sei p_s die Anzahl der primären Knoten mit der Tiefe s, für die gilt
>
> $$Reg(v) \cap L \neq \phi$$
>
> Dann gilt $p_0 = 1$ da $Reg(Wurzel) = R^2$, $p_1 \leq 2$ und
>
> $$p_{s+2} \leq 2p_s$$
>
> Denn von den 4 primären Enkelregionen von v, können höchstens 2 wiedervon L geschnitten werden.



Aus den Rekursionsgleichungen erhält man

$$\begin{array}{ll} p_0 = 1 & \\ p_2 \leq 2 & p_1 \leq 2 \\ p_4 \leq 4 & p_3 \leq 4 \\ p_6 \leq 8 & p_5 \leq 8 \\ p_8 \leq 16 & p_7 \leq 16 \\ \ldots & \ldots \end{array}$$

und allgemein $p_s \leq 2^{\lceil \frac{s}{2} \rceil}$