

Binære søgetræer datastruktur, Består af noder, som udover en key har attributter(left, right, parent) og noget satellitdata - universel pointer til root

Binary search tree property

Balancerede har højst en højde $\log(n)$ fra root til bund simple path - binært kan være lort.

Balancerede gør at basic operationer (search, predecessor, successor, minimum, maximum, insert, delete) tager $O(\log n)$ - ikke for $O(n)$ tid.

Vis procedure for search/access, insert og delete. Make er trivial

Gode hvis højden er lille.

Rød-sorter træer er balancerede binære træ med en ekstra attribut (farve) 1 bit.

Regler for rød-sorter træer: Alt har en farve, roden er sort, leafs er sorte, en node er rød er begge børn sorte, samme antal sorte noder ned.

Hvorfor rødsorte træer er gode søgetræer $O(\log n)$ med lemma-

Lemma 13.1 (rødsort træ har en højde på max $2 \lg(n+1)$)

Proof: Induktion at et subtree med node x har $2^h(x) - 1$ (claim)

Basis: $h = 0$, is a leaf with no nodes.

Vis sandt for $h < k(\text{højden})$

Forestil dig nu subtree med 2 børn.

Rød-sorter træer med rotering (rotationerne). Basic funktioner fungerer på samme måde og opretholder properties ved rotationer og omfarvning op igennem træet.