

Datastruktur: AVL Tree

Oversigt

AVL-træet er en smart udvidelse af det almindelige binære søgetræ. Det holder automatisk sig selv i balance ved at justere sin struktur hver gang der tilføjes eller fjernes elementer. Det smarte er, at ingen gren bliver mere end en højere end dens søskende, som gør det til et AVL-træ.

Data Structure

Et AVL-træ består af følgende komponenter:

- Node: Indeholder data, referencer til venstre og højre child, parent reference, samt højde
- Root: Den øverste node i træet. Har ingen parent
- Parent: En reference til nodens forælder
- Left/Right: Referencer til venstre og højre child
- Height: Nodeens højde, bruges til at beregne balance

Operation	Beskrivelse	Tidskompleksitet
add(item)	Tilføjer et nyt element og rebalancerer træet	$O(\log n)$
remove(item)	Fjerner et element og rebalancerer træet	$O(\log n)$
contains(item)	Søger efter et element i træet	$O(\log n)$
first()/last()	Pinder mindste/største element i træet	$O(\log n)$
traverse()	Gennemløber alle elementer i sorteret rækkefølge	$O(n)$
rebalance()	Genopretter AVL balance egenskaben	$O(1)$
size()	Returnerer antallet af elementer i træet	$O(1)$