

Kjøretider for operasjoner på søkestrukturer

Operasjon	Sortert ArrayList	Naiv binær søketre	AVL Tre	Hashtabell
Innsetting	Bast case: $O(\log(n))$ Average case: $O(n)$ Worst case: $O(n)$	$O(\text{høyden til treet})$	$O(\log(n))$	Best case: $O(1)$ amortized Average case: $O(1)$ amortized Worst case: $\Theta(n)$
Sletting	Bast case: $O(\log(n))$ Average case: $O(n)$ Worst case: $O(n)$	$O(\text{høyden til treet})$	$O(\log(n))$	Best case: $\Theta(1)$ Average case: $\Theta(1)$ Worst case: $\Theta(n)$
Finn element	$O(\log(n))$	$O(\text{høyden til treet})$	$O(\log(n))$	Best case: $\Theta(1)$ Average case: $\Theta(1)$ Worst case: $\Theta(n)$
Finn minste / største	$\Theta(1)$	$O(\text{høyden til treet})$	$O(\log(n))$	Støttes ikke. $O(\text{tabellstørrelse})$ for sekvensielt søk
Finn k-ende minste / største (Selection problemet)	$\Theta(1)$	$O(k * \text{høyden til treet})$	$O(k * \log(n))$ basis $O(\log(n))$ med lagring av størrelse i nodene	Støttes ikke. Kan hente ut nøklene som liste og gjøre QuickSelect på den $O(\text{tabellstørrelse})$.
Finn element over / under (med søk)	$O(\log(n))$	$O(\text{høyden til treet})$	$O(\log(n))$	Støttes ikke. $O(\text{tabellstørrelse})$ for sekvensielt søk
Finn neste element over/under (har posisjonen til elementet)	$\Theta(1)$	$O(\text{høyden til treet}), O(1)$ amortized	$O(1)$ amortized, $O(\log(n))$ worst case	Støttes ikke. $O(\text{tabellstørrelse})$ for sekvensielt søk
Iterer over hele tabellen	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(\text{tabellstørrelse})$