

Datastrukturer – tidskompleksitet

Grid

	første	sidste	midterste	i'te	næste ²
Læs et element ¹	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Find element ³	eksisterer usortet liste	eksisterer sorteret liste	eksisterer ikke usortet liste	eksisterer ikke sorteret liste	
	$O(n)$	$O(\log n)$	$O(n)$	$O(\log n)$	
Indsæt nyt element	i starten	i slutningen	i midten	efter node	før node
	$O(1)^*$	$O(1)^*$	$O(1)^*$	n/a	n/a
Fjern element	første	sidste	i'te	efter node	før node
	$O(1)^*$	$O(1)^*$	$O(1)^*$	n/a	n/a
Byt om på to elementer	første og sidste	første og i'te	sidste og i'te	i'te og j'te	nodes
	$O(1)$	$O(1)$	$O(1)$	$O(1)$	n/a

*hvis vi antager at grid er fast størrelse. Altså at der ikke kan tilføjes/fjernes rækker/kolonner, men blot skrives til hvert index.

¹ At læse et element er som regel det samme som at skrive nyt indhold i et eksisterende element

² Hvis vi allerede har fat i ét element i en datastruktur, kan vi måske læse det "næste" hurtigere end $i+1$ 'te

³ Find et element med en bestemt værdi – alt efter om vi ved at listen er sorteret eller ej, og om elementet findes eller ej.