

Datastrukturer – tidskomplexitet

Double Linked List

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

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 sortet eller ej, og om elementet findes eller ej.