

Datastrukturer – tidskomplexitet

DoublyLinkedList

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

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.