

# Singly Linked List – tidskomplexitet

Skemaer – til sammenligning

## Datastrukturnavn

Læs et element <sup>1</sup>	første	sidste	midterste	i'te	næste <sup>2</sup>
	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$
Find element <sup>3</sup>	eksisterer <i>usorteret liste</i>	eksisterer <i>sorteret liste</i>	eksisterer ikke <i>usorteret 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
	$O(1)$	$O(n)$	$O(n)$		
Fjern element	første	sidste	i'te		
	$O(1)$	$O(n)$	$O(n)$		
Byt om på to elementer	første og sidste	første og i'te	sidste og i'te	i'te og j'te	nodes
	$O(n)$	$O(n)$	$O(n)$	$O(n)$	

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

<sup>2</sup> Hvis vi allerede har fat i ét element i en datastruktur, kan vi måske læse det ”næste” hurtigere end i+1'te

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