

# Array – tidskompleksitet

Et array er den simplest mulige datastruktur

Et array har altid en fast størrelse, elementer kan hverken tilføjes eller fjernes

Elementer ligger altid i en fast rækkefølge, og kan ikke flyttes men ”pladserne” kan overskrives med andre værdier.

## Array

	første	sidste	midterste	i'te	næste <sup>2</sup>
Læs et element <sup>1</sup>	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(n)$ eller $O(1)$ hvis vi kender index
Find element <sup>3</sup>	eksisterer <i>usortet liste</i>	eksisterer <i>sorteret liste</i>	eksisterer ikke <i>usortet liste</i>	eksisterer ikke <i>sorteret liste</i>	
	$O(n)$	$O(\log n)$	$O(n)$	$O(\log n)$	
Indsæt nyt element	i starten	i slutningen	i midten		
	$N/A$	$N/A$	$N/A$		
Fjern element	første	sidste	i'te		
	$N/A$	$N/A$	$N/A$		
Byt om på to elementer	første og sidste	første og i'te	sidste og i'te	i'te og j'te	
	$O(1)$	$O(1)$	$O(1)$	$O(1)$	

<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.