

Array - tidskompleksitet

Skemaer – til sammenligning

Skriv noget klogt her!

Arrays er den simpleste form af en "liste".

De kan ikke ændre størrelse, de har en fast længde.

Vi kan overskrive værdier, men aldrig tilføje eller fjerne elementer.

Array

	første	sidste	midterste	i'te	næste ²
Læs et element ¹	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(n)$ $O(1)$ *
Find element ³	eksisterer usorteret liste	eksisterer sorteret liste	eksisterer ikke usorteret 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
	n/a	n/a	n/a	n/a	n/a
Fjern element	første	sidste	i'te	efter node	før node
	n/a	n/a	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	nodes
	$O(1)$	$O(1)$	$O(1)$	$O(1)$	n/a

Disse scenarios går ud fra worst case.

*, hvis vi kender index

**, vi kan være heldige at finde den før vi har været hele arrayet igennem

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.