

# Datastrukturer – tidskomplexitet

## Queue

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

\*) Hvis man vil læse en kø, læser man som regel alle  $n$  elementer, hvilket tager  $O(n)$  - der er sjældent nogen grund til at læse det  $i$ 'te