

Datastrukturer – tidskomplexitet

Stack

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

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.