

Datastrukturnavn: Queue

Queue (dvs. en kø), implementeret som en linked list med head og tail. Hver node har en pointer til næste node.

Den eneste måde at tilføje elementer på, er ved at placere dem bagerst i køen (ved tail).

Den eneste måde at fjerne elementer på, er ved at fjerne dem fra forrest i køen (ved head).

| | første | sidste | midterste | i'te | |
|---------------------------|--------------------------|---------------------------|-------------------------------|--------------------------------|-------|
| Læs et element | $O(1)$ | $O(1)$ | $O(n)^a$ | $O(n)^b$ | |
| Find element ^c | eksisterer usortet liste | eksisterer sorteret liste | eksisterer ikke usortet liste | eksisterer ikke sorteret liste | |
| | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ | |
| Indsæt nyt element | i starten | i slutningen | i midten | | |
| | n/a | $O(1)$ | n/a | | |
| Fjern element | første | sidste | i'te | | |
| | $O(1)$ | 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^d | n/a | n/a | n/a | |

a) $O(n/2)$

b) $O(i)$

c) Bemærk, at binary search ikke fungerer, for binary search kræver at det tager $O(1)$ at tilgå det i'te element.

d) I en vis forstand kan man godt bytte om på første og sidste, ved at tage det første element ud og placere det bagerst, hvilket tager $O(1)$.