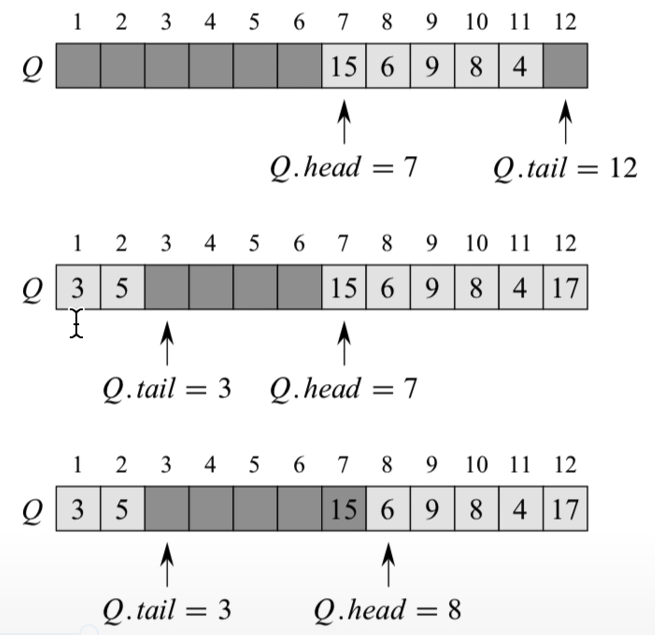
**Queues**

En kø-struktur er baseret på **First-in, first-out** som kan ses som en kø af kunder i en butik. Indsættelse af elementer sker i halen (**tail**) af queuen og sletning sker i hovedet (**head**) af køen.

Køer er implementeret i et array A[1…n] hvor der *max kan værer n-1 elementer.* Indsættelse og sletning hedder *Enqueue* og *dequeue*. En kø Q[1..n] har 2 attributer; **Q.head** og **Q.tail** Hvor head er det ældste element(det som har ventet længst) og tail er ”starten af køen”. **Head og tail er kun pointers til indexe, ikke elementet i det index.** En kø-struktur føler ordenen:

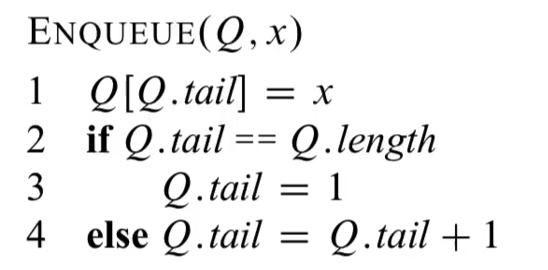
Q.head, Q.head + 1,..., Q.tail – 1. Dette er brugt til at ”wrap around”.

**Eksempel:**

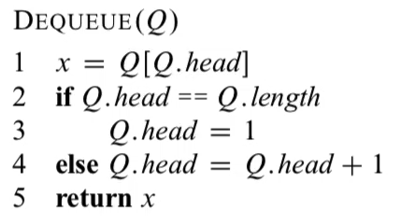


*Elementer som ikke er i index fra head til tail er ikke med i køen. De kan godt have værdier, men ikke en del af køen.*

**Algoritmer**



Enqueue indsætter et element x i en kø. Her bliver elementet indsat og Q.tail bliver inkrementet. Hvis Q.tail = Q.length, så sætter vi Q.tail til 1 (pga wrap around ideen, så der ikke ser et overflow i arrayet).



Dequeue fjerner et element i køen Q. Elementet i Q.head bliver fjernet og Q.head bliver opdateret ligesom i Enqueue. Til sidst returnere den det fjernet element. Ingen af algoritmerne tænker på underflow og overflow i køen. Kørertiden af begge algoritmer er Theta(1) – konstant tid.