

Estructuras Colas

Queues

Estructura Cola (FIFO)

- La cola (queue) es también una estructura lineal restrictiva en la que solo podemos poner y sacar elementos respetando la siguiente restricción:
- El primer elemento en llegar a la cola será también el primero en salir.

FIFO – First In First Out

- Para comprender este concepto, basta con imaginar la cola que se forma en la caja de un supermercado donde la cajera atiende a los clientes que forman la cola respetando el orden de llegada; por lo tanto, el primero que llegó también será el primero en ser atendido y en salir.

- Una cola es otro tipo especial de lista en la cual los elementos se insertan en un extremo (el posterior) y se suprimen en el otro (el anterior o frente).
- Las operaciones para las colas son análogas a las de las pilas; las diferencias sustanciales consisten en que las inserciones se hacen al final de la lista, y no al principio.

Operaciones con colas

1. `Anula(C)` convierte la cola `C` en una lista vacía.
2. `Frente(C)` es una función que devuelve el valor del primer elemento de la cola `C`.
3. `Pone_En_Cola(x, C)` inserta el elemento `x` al final de la cola `C`.
4. `Quita_De_Cola(C)` suprime el primer elemento de `C`.
5. `Vacia(C)` devuelve verdadero si, y solo si, `C` es una cola vacía.



Implementación Estática

Colas

Queues

Implementación Estática – Colas

- Usaremos un simple arreglo para implementar una cola estática.
- Llamamos a la operación insertar un elemento en una cola ENQUEUE
- Llamamos DEQUEUE a la operación eliminar un elemento de la cola.
- La cola tiene cabeza y cola.

- La propiedad FIFO de una cola hace que funcione como una línea de clientes esperando pagarle a un cajero.
- Cuando un elemento se pone en cola, ocupa su lugar al final de la cola, al igual que un cliente recién llegado ocupa un lugar al final de la fila.
- El elemento quitado de la cola es siempre el que está en la cabecera de la cola, como el cliente en la cabecera de la fila que ha esperado el tiempo más largo.

- La Figura 1 muestra una forma de implementar una cola de como máximo **$n-1$** elementos utilizando un arreglo **$Q[1..n]$** .
- La cola tiene un atributo **$Q.head$** que indexa, o apunta a su cabeza.
- El atributo **$Q.tail$** indexa a la siguiente ubicación en la que un elemento recién llegado se insertará en la cola.
- Los elementos de la cola residen en las ubicaciones **$Q.head$** , **$Q.head+1$** , ..., **$Q.tail-1$** , donde se entiende que la ubicación **1** sigue inmediatamente después de la ubicación **n** en un orden circular.

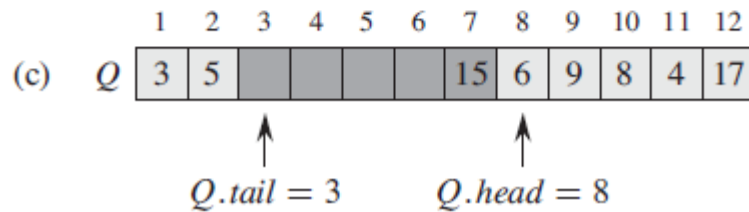
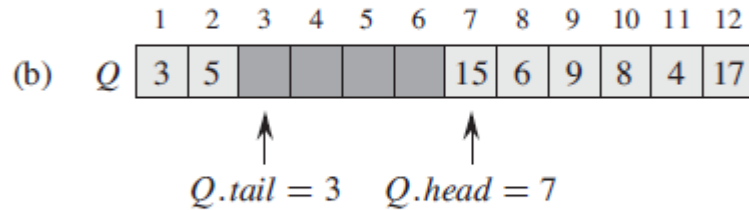
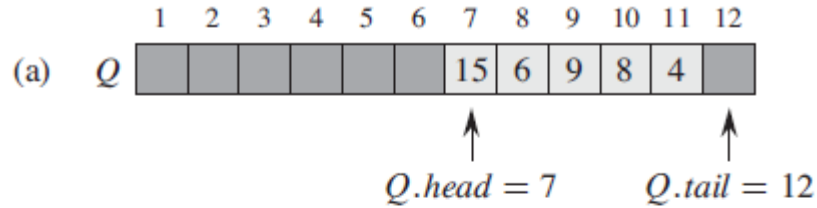


Figura 1. Una cola implementada usando un arreglo $Q[1...12]$.

Los elementos de la cola aparecen solo en las posiciones ligeramente sombreadas.

A. La cola tiene 5 elementos, en las ubicaciones $Q[7...11]$.

B. La configuración de la cola después de las llamadas:

`ENQUEUE(Q,17)`

`ENQUEUE(Q, 3)`

`ENQUEUE(Q, 5)`

C. La configuración de la cola después de la llamada `DEQUEUE(Q)` devuelve el valor de 15 que esta formado en la cabeza de la cola.

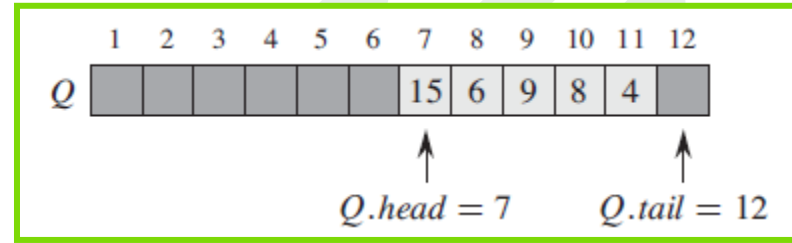
La nueva cabeza tiene el valor de 6.

- La **cola está vacía** cuando se cumple que:

$$Q.head = Q.tail$$

Inicialmente, tenemos $Q.head = Q.tail = 1$.

Si intentamos sacar un elemento de una cola vacía, la cola se desborda.



- La **cola está llena** cuando:

$$Q.head = Q.tail + 1$$
$$Q.head = 1 \text{ y } Q.tail = Q.length$$

Si intentamos poner en cola un elemento, entonces la cola se desborda.

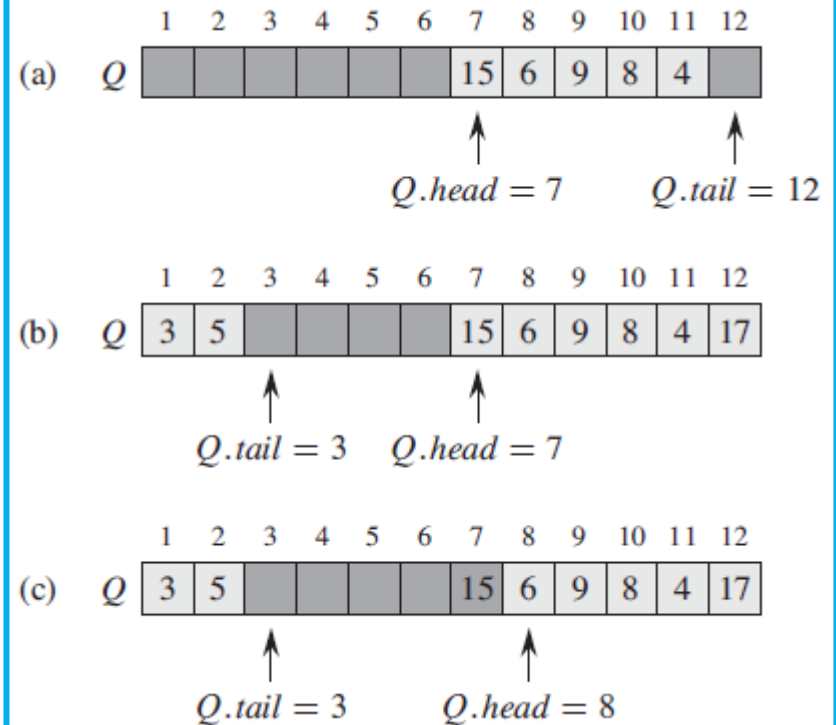
- El pseudocódigo asume que $n = Q.length$

ENQUEUE(Q, x)

```
1   $Q[Q.tail] = x$ 
2  if  $Q.tail == Q.length$ 
3       $Q.tail = 1$ 
4  else  $Q.tail = Q.tail + 1$ 
```

DEQUEUE(Q)

```
1   $x = Q[Q.head]$ 
2  if  $Q.head == Q.length$ 
3       $Q.head = 1$ 
4  else  $Q.head = Q.head + 1$ 
5  return  $x$ 
```



- En nuestros procedimientos ENQUEUE y DEQUEUE, hemos omitido la comprobación de errores para:
 - Subdesbordamiento (cuando se quiere sacar un elemento y la cola esta vacía)
 - Desbordamiento (cuando se quiere agregar un elemento y la cola esta llena).