

Guía práctica de estudio 05.

Estructuras de datos lineales: Pila y cola.



Elaborado por:

M.C. Edgar E. García Cano
Ing. Jorge A. Solano Gálvez

Autorizado por:

M.C. Alejandro Velázquez Mena

Guía práctica de estudio 05.

Estructuras de datos lineales: Pila y cola.

Objetivo:

Revisarás las definiciones, características, procedimientos y ejemplos de las estructuras lineales Pila y Cola, con la finalidad de que comprendas sus estructuras y puedas implementarlas.

Actividades:

- Revisar definición y características de la estructura de datos pila.
- Revisar definición y características de la estructura de datos cola.
- Implementar las estructuras de datos pila y cola.

Introducción

Los conjuntos (colecciones de datos) son tan fundamentales para las ciencias de la computación como lo son para las matemáticas.

Una estructura de datos consiste en una colección de nodos o registros del mismo tipo que mantienen relaciones entre sí. Un nodo es la unidad mínima de almacenamiento de información en una estructura de datos.

Las estructuras de datos lineales son aquellas en las que los elementos ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y un único predecesor.

Pila

La pila (o stack) es una estructura de datos lineal y dinámica, en la cual el elemento obtenido a través de la operación ELIMINAR está predefinido, debido a que implementa la política Last-In, First-Out (LIFO), esto es, el último elemento que se agregó es el primer que se elimina.

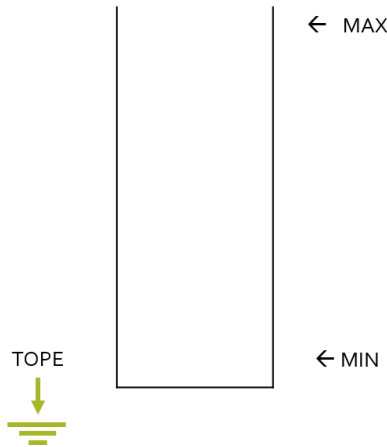
Las operaciones que se pueden realizar sobre una pila son INSERTAR (que es llamada PUSH) y ELIMINAR (que es llamada POP). Debido a la política LIFO que implementa esta estructura, el orden en el que los elementos son extraídos de la pila (POP) es inverso al orden en el que los elementos fueron insertados en la pila (PUSH). Además, el único elemento accesible de la pila es el que está hasta arriba y que se conoce como *tope* de la pila.

Para poder diseñar un algoritmo que defina el comportamiento de una pila se deben considerar 3 casos para ambas operaciones (push y pop):

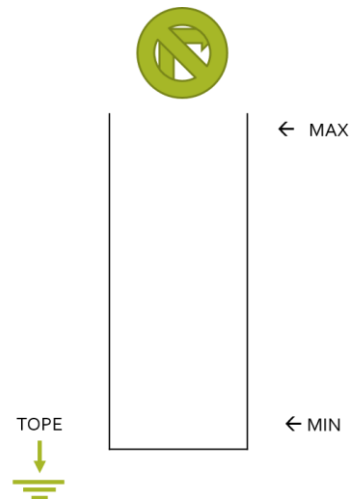
- Estructura vacía (caso extremo).
- Estructura llena (caso extremo).
- Estructura con elemento(s) (caso base).

Pila vacía

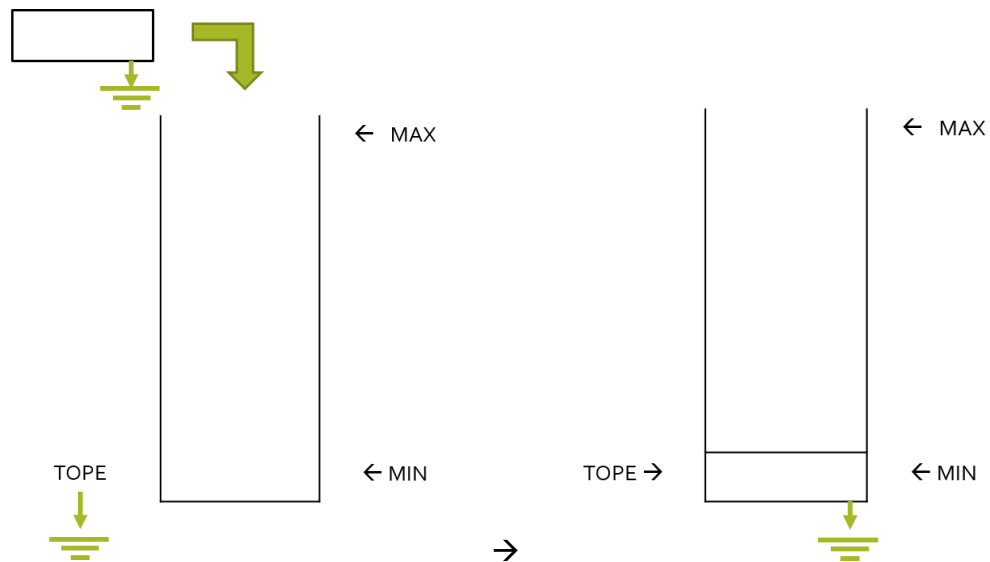
Una pila vacía no contiene elemento alguno dentro de la estructura y el *tope* de la misma apunta a *nulo*.



En una pila vacía no es posible realizar POP, debido a que la estructura no contiene información.

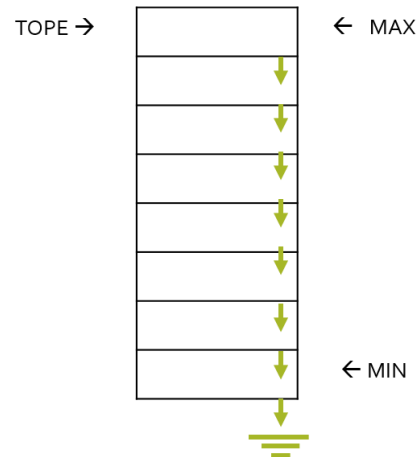


Cuando la pila está vacía sí se puede realizar PUSH, en tal caso, el nodo que entra a la estructura sería el único elemento de la pila y el *tope* apuntaría a él.

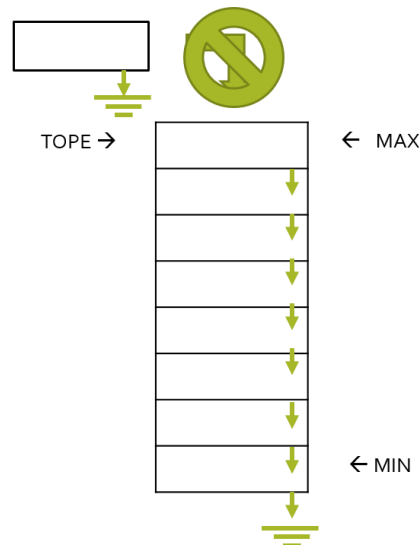


Pila llena

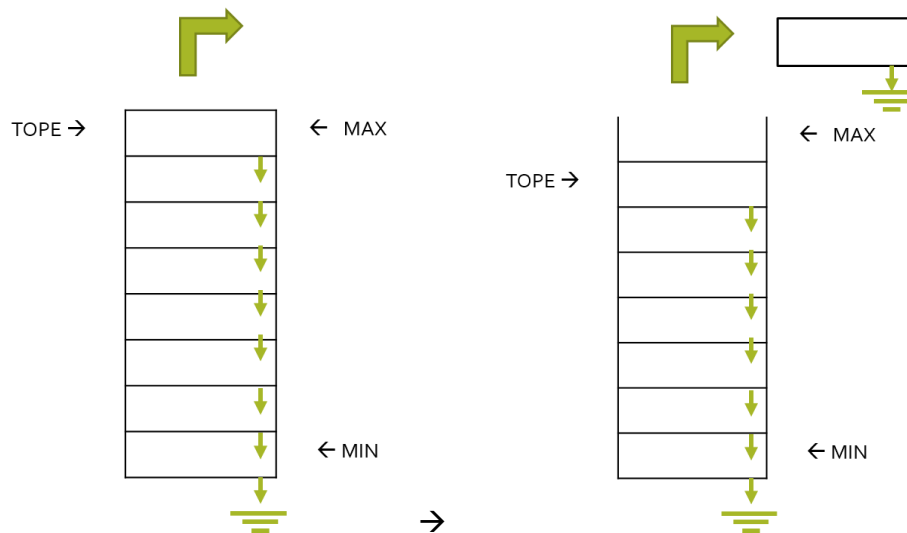
Por definición, una estructura de datos tipo pila tiene un tamaño fijo. Cuando la pila ha almacenado el número máximo de nodos definido, se dice que la pila está llena.



En una pila llena no es posible hacer PUSH de un nuevo elemento, ya que se ha alcanzado el tamaño máximo permitido.

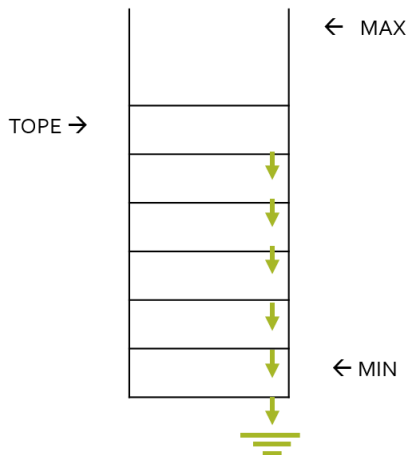


Cuando la pila está llena se puede hacer POP de la información contenida en la estructura. En tal caso, el *tope* apunta al elemento siguiente de la estructura.

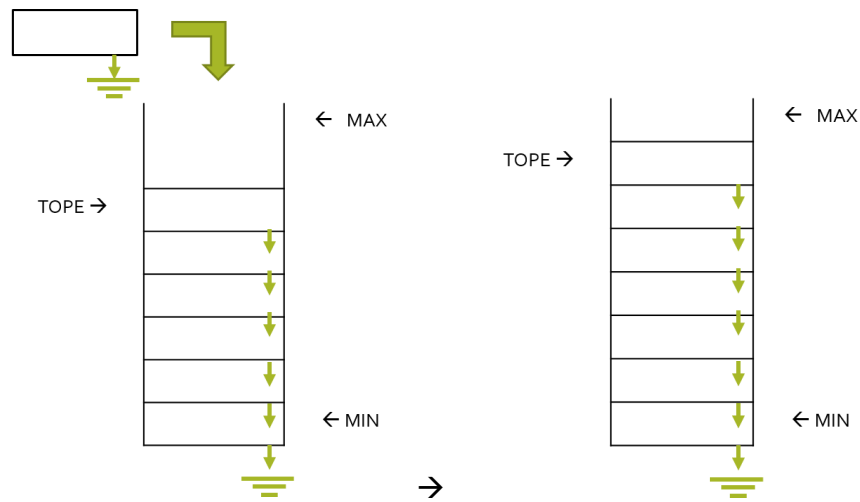


Pila con elementos

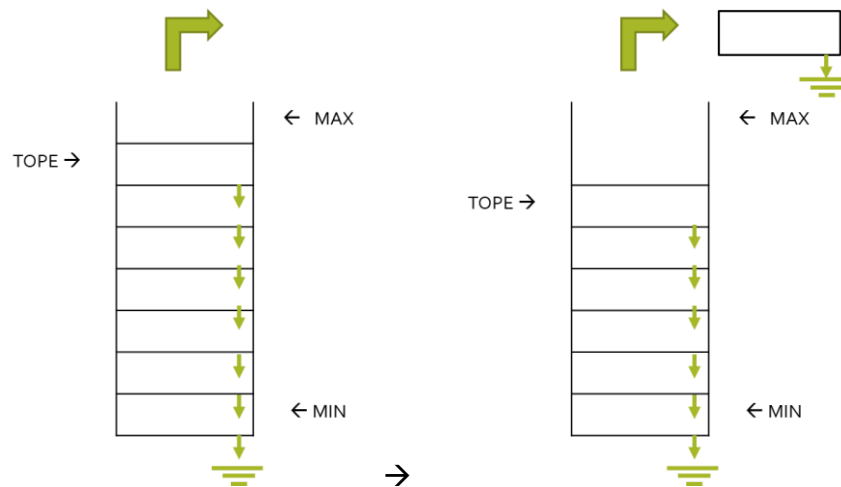
Una pila que contiene elementos (sin llegar a su máxima capacidad) representa el caso general.



En una pila con elementos se pueden realizar PUSH. En tal caso, el *tope* apuntara al elemento que se insertó y el nuevo elemento apunta al elemento al que apuntaba *tope*.



En una pila con elementos es posible realizar POP. En tal caso, el nodo al que apunta *tope* se extrae y ahora *tope* apunta al elemento al que apuntaba éste (sucesor).



Aplicaciones

La estructura pila tienen varias aplicaciones dentro de la ingeniería, de las más conocidas es la que se utiliza dentro de la memoria RAM de un equipo de cómputo.

La memoria de las computadoras no es un espacio uniforme, el código que se ejecuta utiliza tres diferentes segmentos de memoria: el texto (text), la pila (stack) y el montículo (heap)

Cuando una aplicación inicia, el método principal es invocado y se reserva memoria en la pila o stack. En el segmento de memoria de la pila es donde se alojan las variables requeridas por las funciones del programa. Así mismo, cada vez que se llama una función dentro del programa una sección de la pila, llamada marco o frame, se reserva y es ahí donde las variables de la nueva función son almacenadas.

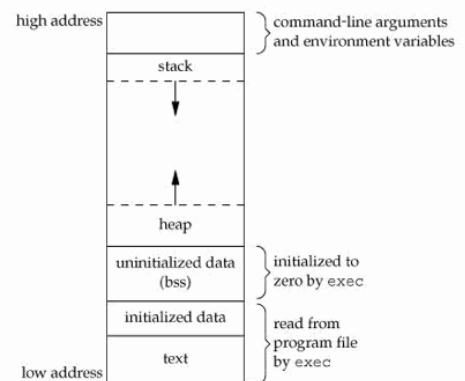


Figura 1. Regiones de la memoria RAM

Cuando una función manda llamar varias funciones, éstas generan un nuevo marco que se va creando uno encima del otro y, cuando las funciones terminan, los marcos se liberan de manera automática en orden inverso (LIFO).

Cola

La cola (o queue) es una estructura de datos lineal, en la cual el elemento obtenido a través de la operación ELIMINAR está predefinido y es el que se encuentra al inicio de la estructura.

La cola implementa la política First-In, First-Out (FIFO), esto es, el primer elemento que se agregó es el primero que se elimina.

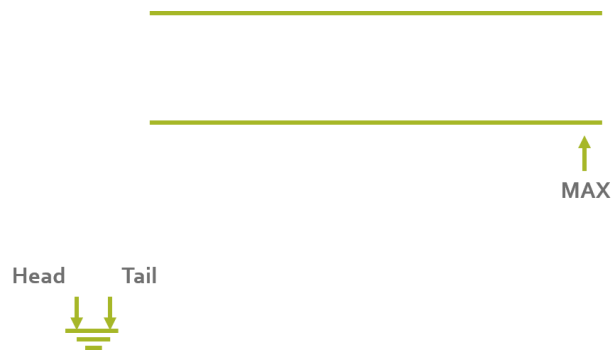
La cola es una estructura de datos de tamaño fijo y cuyas operaciones se realizan por ambos extremos; permite INSERTAR elementos al final de la estructura y permite ELIMINAR elementos por el inicio de la misma. La operación de INSERTAR también se le llama ENCOLAR y la operación de ELIMINAR también se le llama DESENCOLAR.

Para poder diseñar un algoritmo que defina el comportamiento de una COLA se deben considerar 3 casos para ambas operaciones (INSERTAR y ELIMINAR):

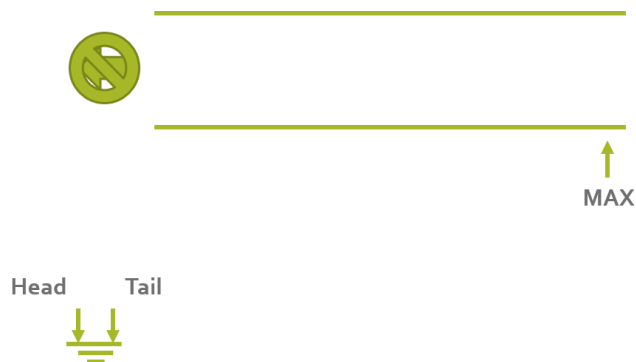
- Estructura vacía (caso extremo).
- Estructura llena (caso extremo).
- Estructura con elemento(s) (caso base).

Cola vacía

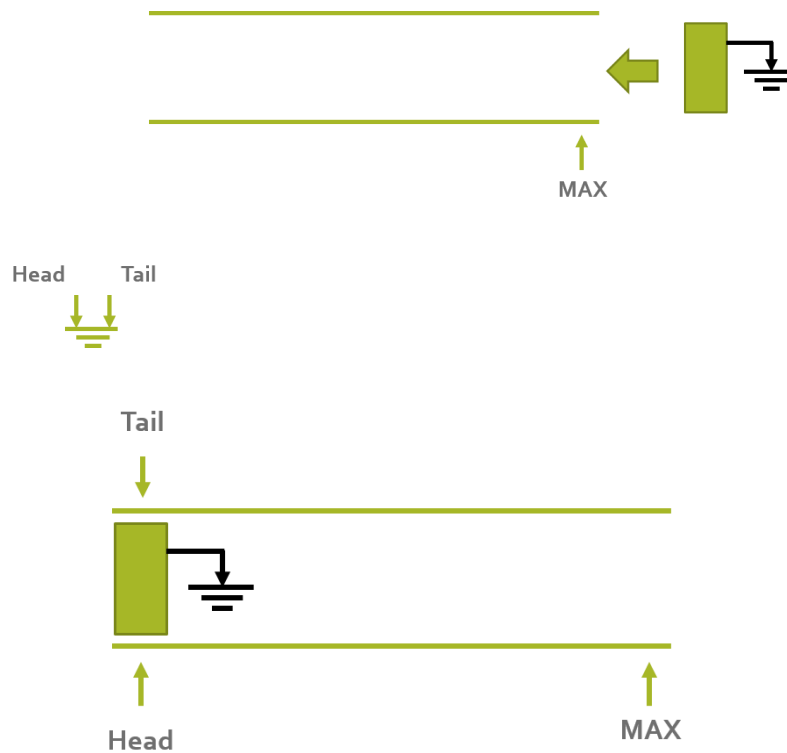
La cola posee dos referencias, una al inicio (HEAD) y otra al final (TAIL) de la cola. En una cola vacía ambas referencias (HEAD y TAIL) apuntan a *nulo*.



En una cola vacía no es posible desencolar debido a que la estructura no posee elementos.

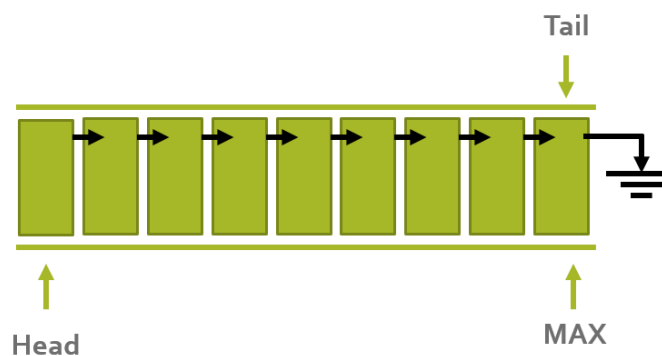


En una cola vacía sí se pueden encolar elementos, en este caso las referencias HEAD y TAIL apuntan al mismo elemento, que es el único en la estructura.

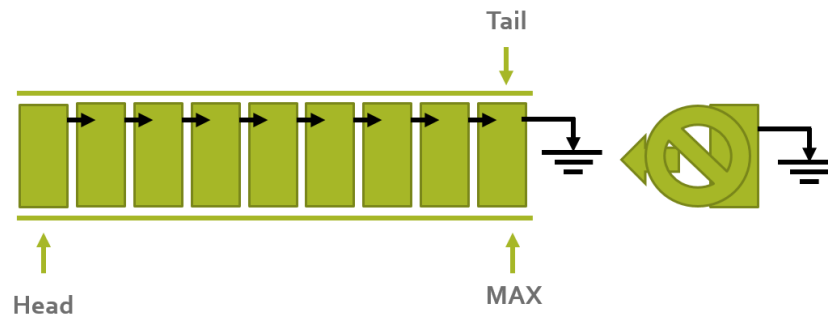


Cola llena

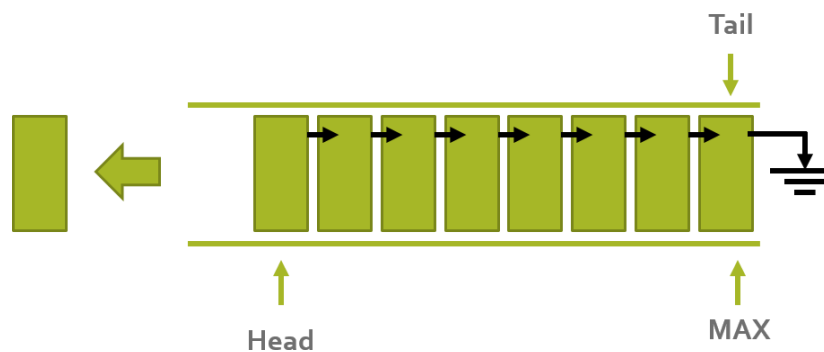
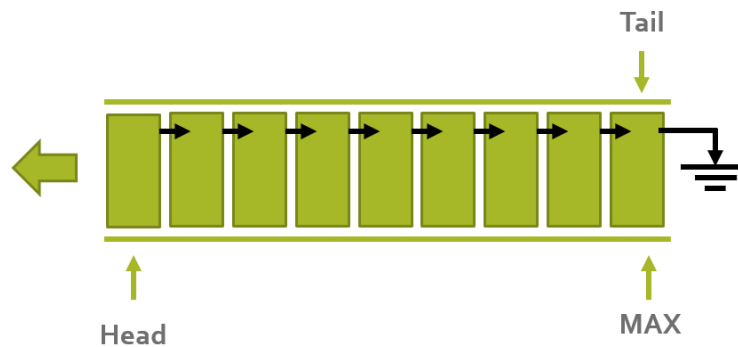
Cuando la referencia a tail de una cola llega a su máxima capacidad de almacenamiento (MAX) se dice que la cola está llena.



En una cola llena no es posible encolar más elementos.

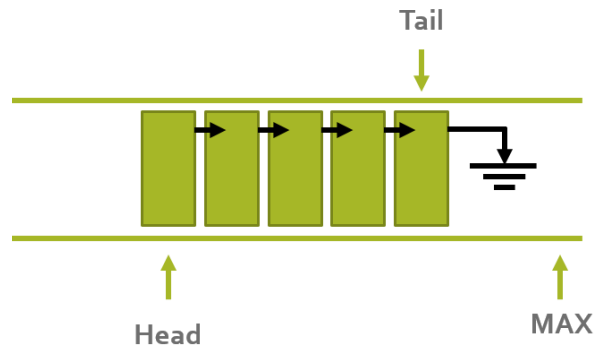


En una cola llena sí se pueden desencolar elementos, en tal caso se obtiene el elemento al que hace referencia head y esta referencia se recorre al siguiente elemento (sucesor).

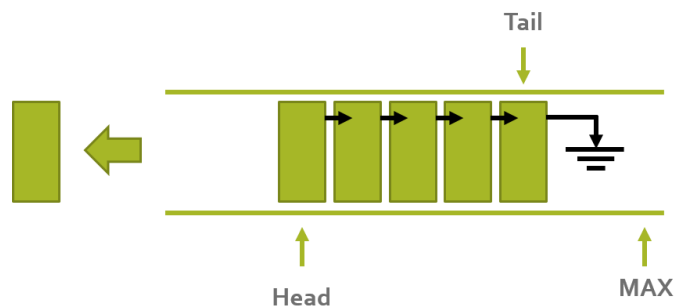
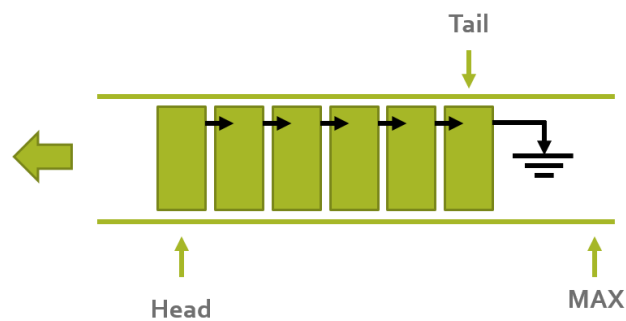


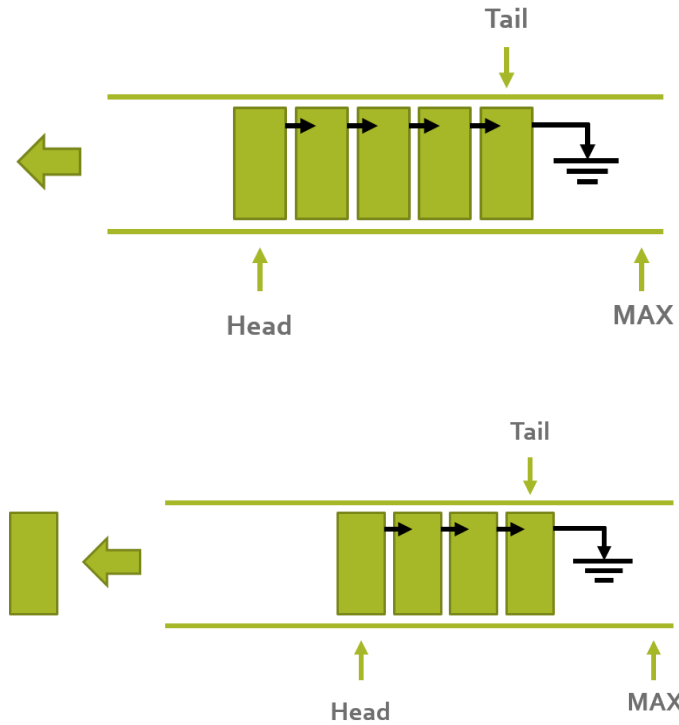
Cola con elementos

Una cola que contiene elementos (sin llegar a su máximo tamaño) representa el caso general de la estructura.

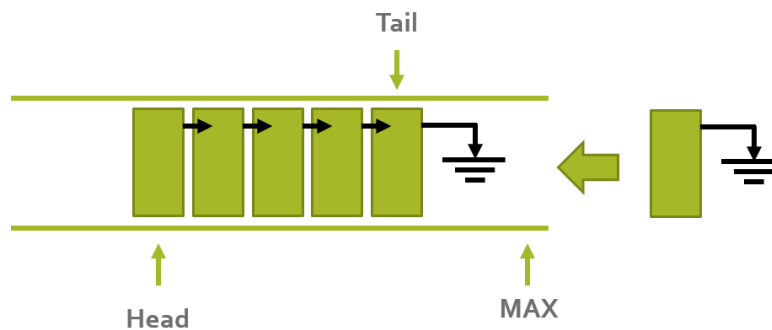


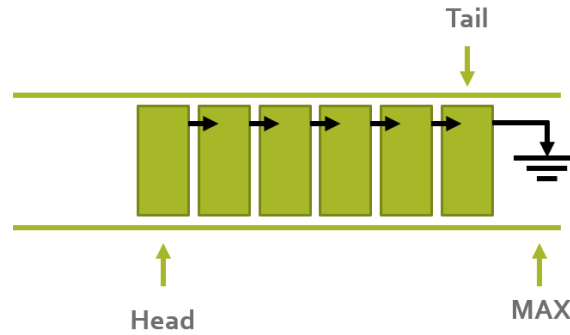
En una cola con elementos es posible desencolar nodos, recorriendo la referencia al inicio de la cola (HEAD) al siguiente elemento de la estructura.





Así mismo, se pueden encolar elementos en una cola mientras la referencia al final (TAIL) de la estructura no sea mayor al tamaño máximo de la misma. Cuando se encola un elemento, el nodo al que apunta TAIL tiene como sucesor el nuevo elemento y la referencia a TAIL apunta al nuevo elemento.





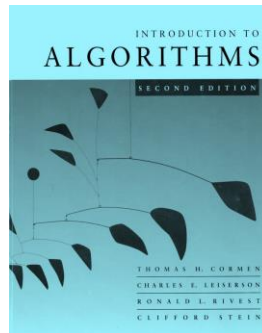
Aplicaciones

La aplicación más conocida de la estructura cola es la que se utiliza en la impresión de documentos.

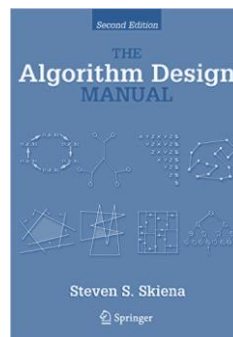
Las impresoras tienen una cantidad de memoria limitada, la cual puede ser inferior al tamaño de un documento que se desea imprimir.

La cola de impresión permite enviar documentos de gran tamaño, o varios documentos, a una impresora sin tener que esperar que se complete la impresión para seguir con la siguiente tarea. Cuando se envía un archivo a imprimir, se crea un archivo de almacenamiento intermedio en formato EMF, donde se almacena lo que se envía a la impresora y las opciones de impresión. Las impresiones se van realizando según vayan llegando los archivos (FIFO).

Bibliografía



Introduction to Algorithms. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, McGraw-Hill.



The Algorithm Design Manual. Steven S. Skiena, Springer.

Ariel Rodríguez (2010). How knowing C and C++ can help you write better iPhone apps, part 1. [Figura 1]. Consulta: Enero de 2016. Disponible en: <http://akosma.com/2010/10/11/how-knowing-c-and-c-can-help-you-write-better-iphone-apps-part-1/>