



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martínez Quintana

Asignatura: Estructura de Datos y Algoritmos 1

Grupo: 15

No de Práctica(s): 6

Integrante(s): Citlali Cuahtepitzi Cuatlapantzi

*No. de Equipo de
cómputo empleado:* NO APLICA

No. de Lista o Brigada: NO APLICA

Semestre: 2021-2

Fecha de entrega: 6 de agosto del 2021

Observaciones: _____

CALIFICACIÓN: _____

Guía práctica de estudio 05- 07. Estructuras de datos lineales: Cola, Lista simple y lista circular

OBJETIVO:

Revisarás las definiciones, características, procedimientos y ejemplos de la estructura lineal Cola, Lista simple y Lista circular, con la finalidad de que comprendas sus estructuras y puedas implementarlas.

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.

Las listas son un tipo de estructura de datos lineal y dinámica. Es lineal porque cada elemento tiene un único predecesor y un único sucesor, y es dinámica porque su tamaño no es fijo y se puede definir conforme se requiera. Las operaciones básicas dentro de una lista son BUSCAR, INSERTAR Y ELIMINAR.

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.

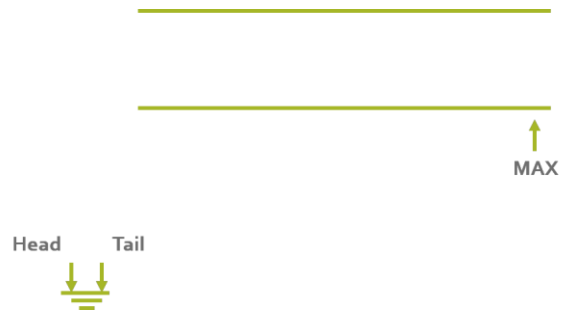
La cola es una estructura de datos de tamaño fijo y cuyas operaciones se realizan por ambos extremos; permite INSERTAR elementos al final de la estructura y permite ELIMINAR elementos por el inicio de la misma. La operación de INSERTAR también se le llama ENCOLAR y la operación de ELIMINAR también se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

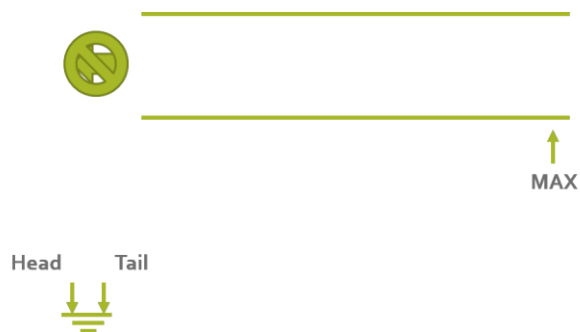
- Estructura vacía (caso extremo).
- Estructura llena (caso extremo).
- Estructura con elemento(s) (caso base).

Cola vacía

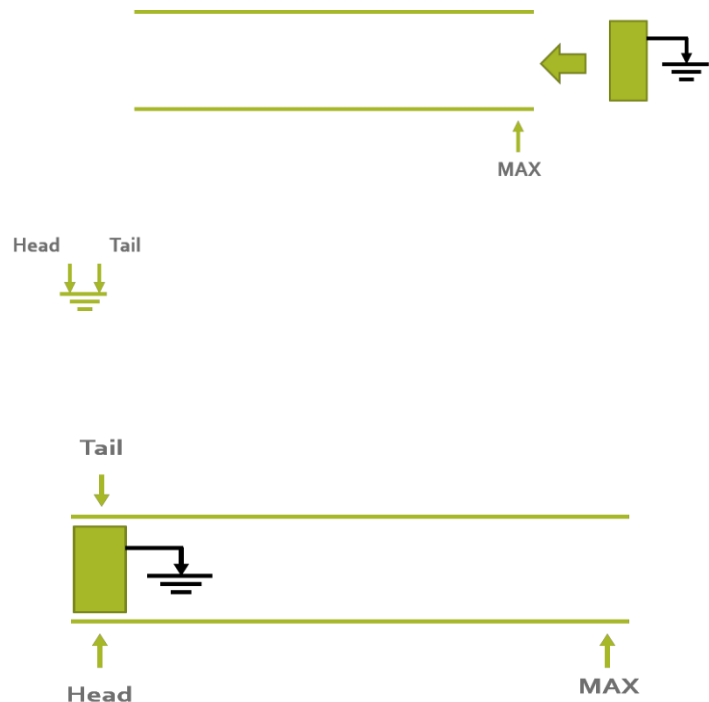
La cola posee dos referencias, una al inicio (HEAD) y otra al final (TAIL) de la cola. En una cola vacía ambas referencias (HEAD y TAIL) apuntan a nulo.



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

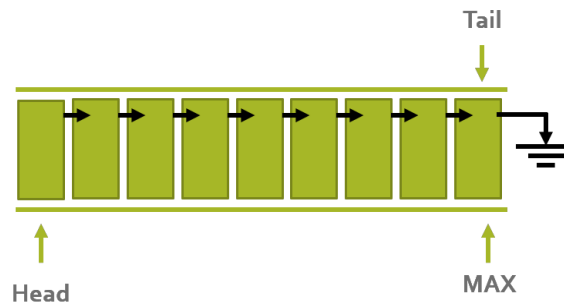


En una cola vacía sí se pueden encolar elementos, en este caso las referencias HEAD y TAIL apuntan al mismo elemento, que es el único en la estructura.

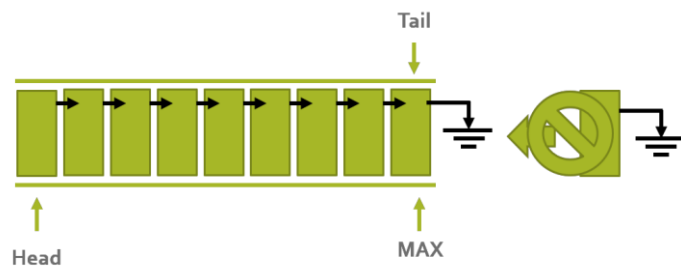


Cola llena

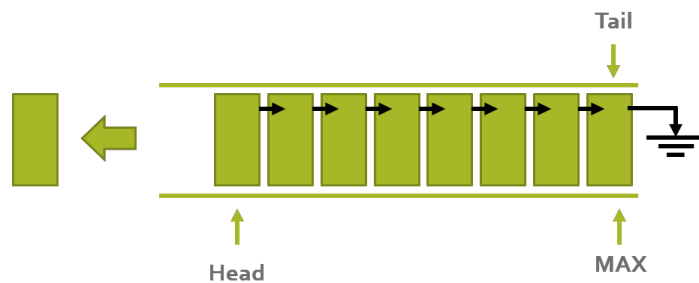
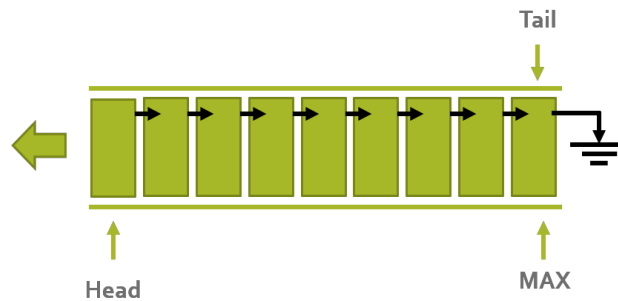
Cuando la referencia a tail de una cola llega a su máxima capacidad de almacenamiento (MAX) se dice que la cola está llena.



En una cola llena no es posible encolar más elementos.

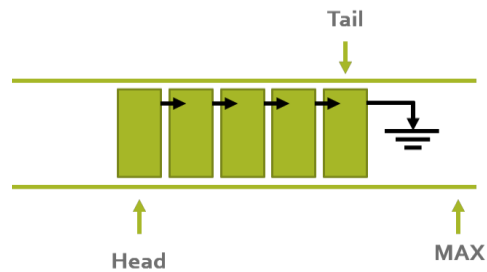


En una cola llena sí se pueden desencolar elementos, en tal caso se obtiene el elemento al que hace referencia head y esta referencia se recorre al siguiente elemento (sucesor).

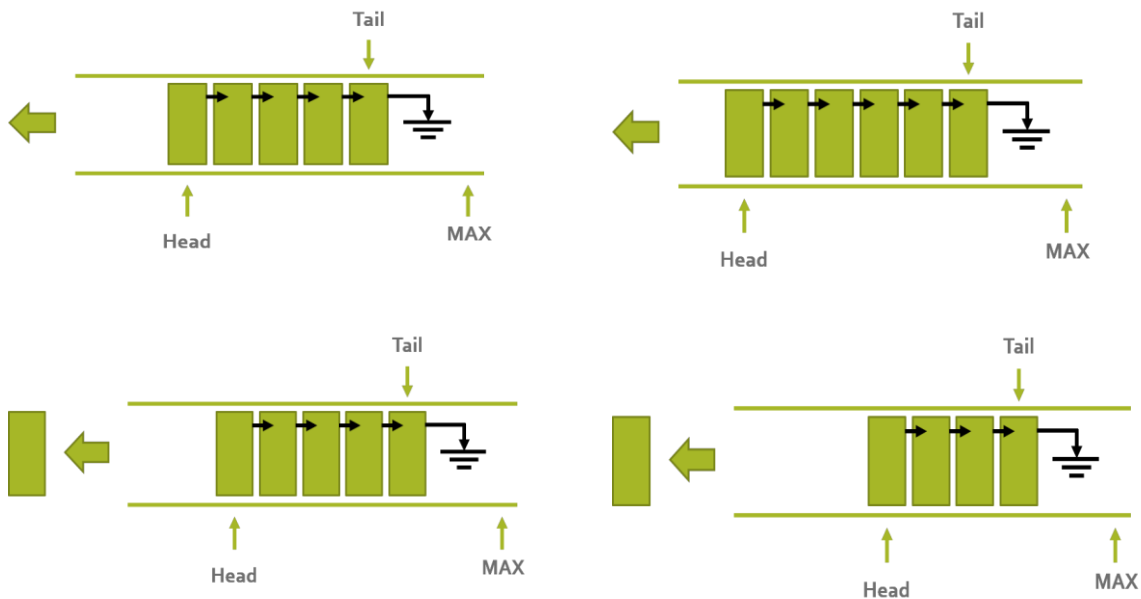


Cola con elementos

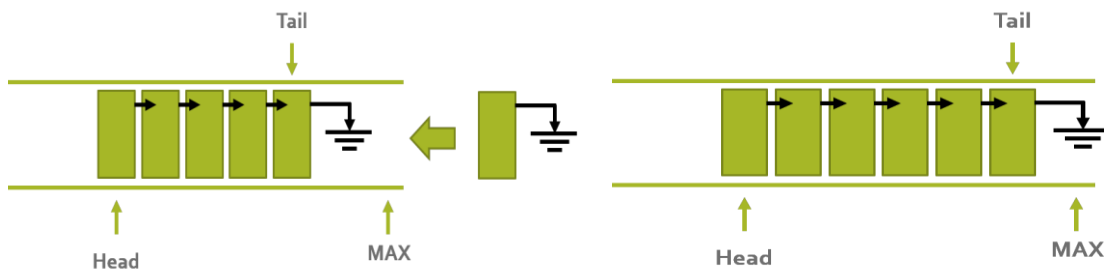
Una cola que contiene elementos (sin llegar a su máximo tamaño) representa el caso general de la estructura.



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.



Lista Simple

Una lista simple (también conocida como lista ligada o lista simplemente ligada) está constituida por un conjunto de nodos alineados de manera lineal (uno después de otro) y unidos entre sí por una referencia.

A diferencia de un arreglo, el cual también es un conjunto de nodos alineados de manera lineal, el orden está determinado por una referencia, no por un índice, y el tamaño no es fijo.

La unidad básica de una lista simple es un elemento o nodo. Cada elemento de la lista es un objeto que contiene la información que se desea almacenar, así como una referencia (NEXT) al siguiente elemento (SUCEsor).



Para poder diseñar un algoritmo que defina el comportamiento de una LISTA LIGADA se deben considerar 2 casos para cada operación (BUSCAR, INSERTAR y ELIMINAR):

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

Buscar

El método debe buscar el primer elemento que coincida con la llave K dentro de la lista L, a través de una búsqueda lineal simple, regresando un apuntador a dicho elemento si éste se encuentra en la lista o nulo en caso contrario.

Una lista simple vacía no contiene elementos, la referencia al inicio de la misma (head) apunta a nulo, por lo tanto, en una lista vacía no es posible buscar elementos.

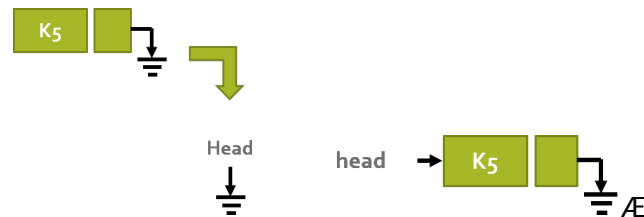


Una lista simple con elementos puede contener de 1 a n elementos, en tal caso, la referencia al inicio (HEAD) apunta al primer elemento de la lista. Es posible recorrer la lista a través de la referencia (NEXT) de cada nodo hasta llegar al que apunta a nulo, el cual será el último elemento. Por lo tanto, dentro de una lista simple con elementos es posible buscar una llave K.

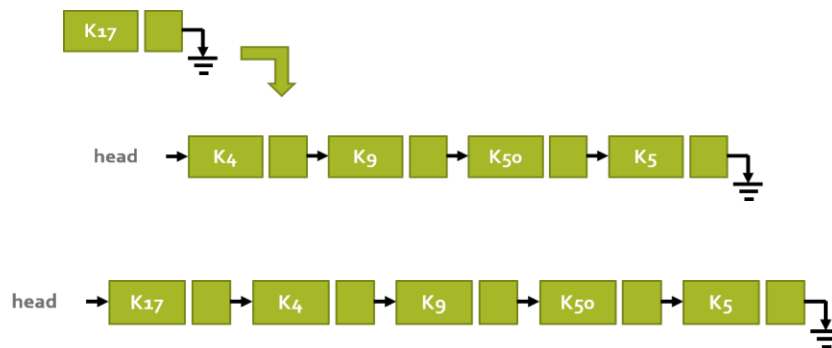


Insertar

Dado un nodo x que contenga una llave k previamente establecida, el método INSERTAR agrega el elemento x al inicio de la lista. Es posible insertar elementos tanto en una lista simple vacía como en una lista simple con elementos. Cuando se inserta un nuevo elemento en una lista simple vacía la referencia al inicio de la lista (HEAD) apunta al nodo insertado.



Cuando se inserta un nuevo elemento en una lista simple con elementos, la referencia del nuevo nodo (NEXT) apunta al mismo nodo al que apunta el inicio de la lista (HEAD) y ahora HEAD apunta al nuevo nodo.



Borrar

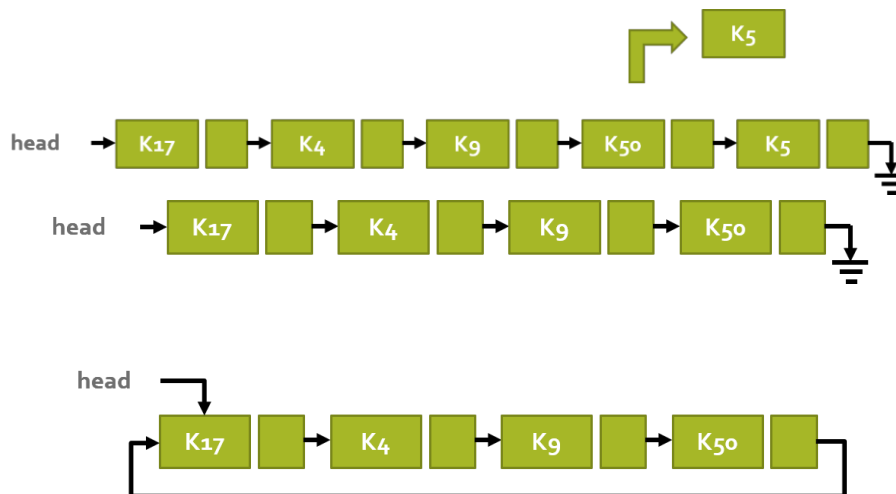
El método elimina el elemento x de la lista L (si es que éste se encuentra en la estructura). Para eliminar un elemento de la lista primero es necesario saber la ubicación del nodo a eliminar, por lo tanto, primero se debe realizar una búsqueda del elemento.

En una lista simple vacía no es posible eliminar, debido a que esta estructura no contiene elementos.

Para eliminar un nodo en una lista simple con elementos, primero se debe buscar el elemento a eliminar, una vez encontrado el nodo en la lista, se deben mover las referencias de la estructura de tal manera de que el antecesor del nodo a eliminar apunte al sucesor del mismo.

Lista circular

Una lista circular es una lista simplemente ligada modificada, donde el apuntador del elemento que se encuentra al final de la lista (TAIL) apunta al primer elemento de la lista (HEAD).



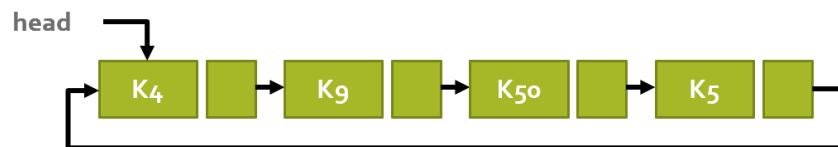
Buscar

El método debe buscar el primer elemento que coincida con la llave K dentro de la lista L, a través de una búsqueda lineal simple, regresando un apuntador a dicho elemento si éste se encuentra en la lista o nulo en caso contrario.

Una lista circular vacía no contiene elementos, la referencia al inicio de la misma (HEAD) apunta a NULO, por lo tanto, en una lista vacía no es posible buscar elementos.



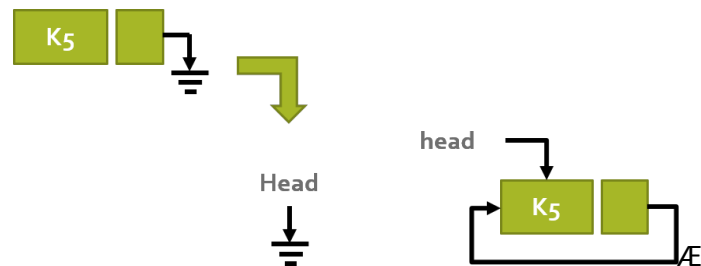
Una lista circular con elementos puede contener de 1 a n elementos, en tal caso, la referencia al inicio (HEAD) apunta al primer elemento de la lista y la referencia a NEXT del último elemento apunta al primer elemento. Es posible recorrer la lista a través de la referencia (NEXT) de cada nodo, hay que tener en cuenta el número de elementos de la lista, ya que el último elemento apunta al inicio de la estructura y, por tanto, se puede recorrer de manera infinita. Dentro de una lista circular con elementos es posible buscar una llave K.



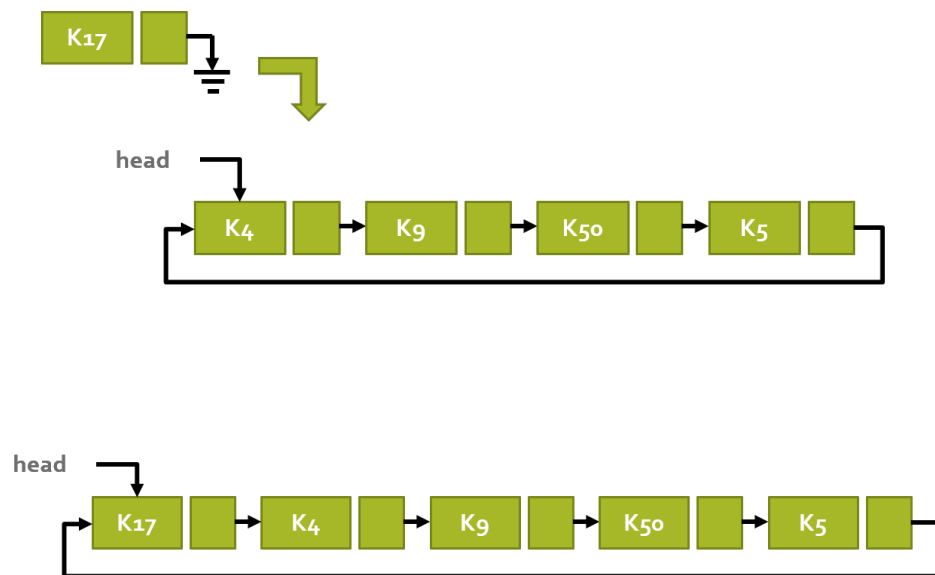
Insertar

Dado un nodo x que contenga una llave K previamente establecida, el método INSERTAR agrega el elemento x al inicio de la lista.

Es posible insertar elementos tanto en una lista circular vacía como en una lista circular con elementos. Cuando se inserta un nuevo elemento en una lista circular vacía la referencia al inicio de la lista (HEAD) apunta al nodo insertado y la referencia a NEXT del nodo apunta a sí mismo.



Cuando se inserta un nuevo elemento en una lista circular con elementos, la referencia del nuevo nodo (NEXT) apunta al mismo nodo al que apunta el inicio de la lista (HEAD) y ahora HEAD apunta al nuevo nodo. Así mismo, el último nodo de la estructura (TAIL) apunta al primer elemento.

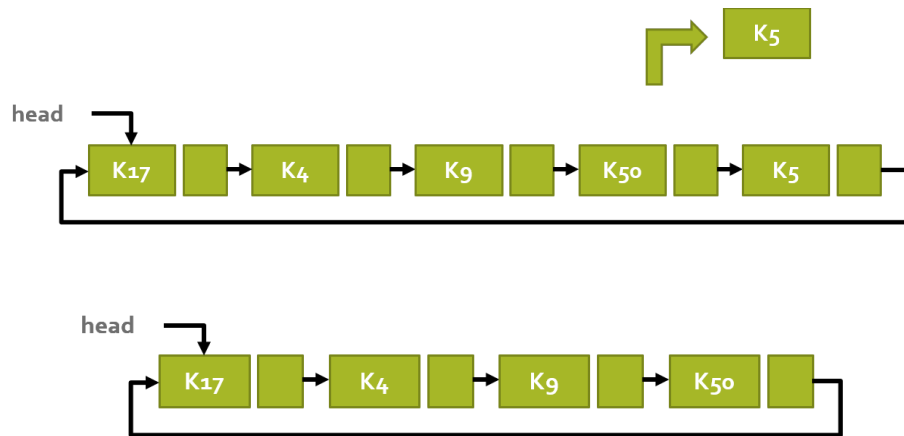


Borrar

El método elimina el elemento x de la lista L (si es que éste se encuentra en la estructura). Para eliminar un elemento de la lista primero es necesario saber la ubicación del nodo a eliminar, por lo tanto, primero se debe realizar una búsqueda del elemento.

En una lista circular vacía no es posible eliminar, debido a que esta estructura no contiene elementos.

Para eliminar un nodo en una lista circular con elementos, primero se debe buscar el elemento a eliminar, una vez encontrado el nodo en la lista, se deben mover las referencias de la estructura de tal manera de que el antecesor del nodo a eliminar apunte al sucesor del mismo.



DESARROLLO

Aplicación de colas

Principalmente son utilizadas en actividades de gestión de recursos, sistemas de tiempo compartido, colas de impresión y simulaciones por computadora de situaciones reales.

- Sistemas de tiempo compartido: los recursos (CPU, memoria, ...) se asignan a los procesos que están en cola de espera en el orden en el que fueron introducidos.
- Colas de impresión: al intentar imprimir varios documentos a la vez o la impresora está ocupada, los trabajos se almacenan en una cola según el orden de llegada.
- Simulación por computadora de situaciones reales: una cola de clientes en un supermercado o el tiempo de espera para ser atendidos por un operador de una línea telefónica.

Aplicaciones de listas

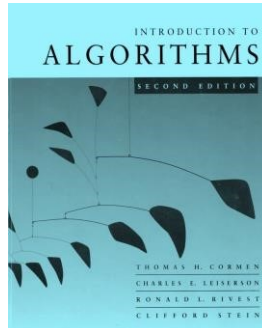
Las listas son una de las estructuras de datos más utilizadas en las ciencias de la computación.

- Cualquier red social utiliza una lista simple, en la que cada elemento tiene un único sucesor que sería la siguiente publicación, hasta llegar a la última.
- En cualquier reproductor de audio, una lista de canciones se puede reproducir de manera ordenada o de manera desordenada (aleatoria). Así mismo, se puede repetir la lista de reproducción de manera automática, es decir, el sucesor del último elemento de la lista es el primer elemento de la lista, lo que genera una lista circular.
- Lo mismo pasa con plataformas de video al generarse listas de reproducción.

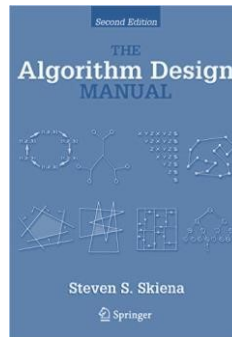
CONCLUSIÓN

Las estructuras de datos lineales están presentes en distintas actividades que realizamos como parte de su funcionamiento, la mayor parte del tiempo no nos detenemos a pensar como es la labor que ejecutan todas estas estructuras desde su centro de datos, algoritmo y programación completa. Es importante comprender las operaciones principales de estas estructuras pues son la base para entender como vivimos en la actualidad, además de que te da el conocimiento para seguir modificando el estilo de vida para bien a base de la programación.

BIBLIOGRAFÍA O REFERENCIAS:



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-betteriphone-apps-part-1/>

Jaqueline, S. E. PILAS Y COLAS. Recuperado de:
<http://hdl.handle.net/20.500.11799/34181>