



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

Ing. Marco Antonio Martínez Quintana

Asignatura:

Estructura de datos y algoritmos I

Grupo:

17

No de Práctica(s):

7

Integrante(s):

Vega Gutierrez Ricardo Daniel

*No. de Equipo de
cómputo empleado:*

casa

No. de Lista o Brigada:

40

Semestre:

2020-2

Fecha de entrega:

17 de marzo del 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo

Se conocerá un nuevo tipo de estructuras líneas; listas simples enlazadas y lista circular, todo con la finalidad de usarlas y aplicarlas a la resolución de problemas.

Introducción

Las listas son estructuras de tipo lineal, debido a que solo tienen un sucesor y antecesor. A su vez también se dice, que son dinámicas porque su tamaño puede ser modificado, dependiendo su utilidad. Las operaciones básicas de las listas son: 1) Buscar, 2) Insertar y 3) Eliminar.

Lista simple

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).



La unidad básica de las listas simples son los nodos, que a su vez cada uno de ellos es un lugar de memoria que contiene información que se desea almacenar, puede ser una referencia (NEXT) que apunte al siguiente elemento o nodo (SUCESOR).

Una lista simple se compone de nodos los cuales son están ligados de manera lineal, o sea uno después del otro, el orden de los mismos se determina por una referencia.

Las listas simples tienen dos casos:

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

Función - Buscar

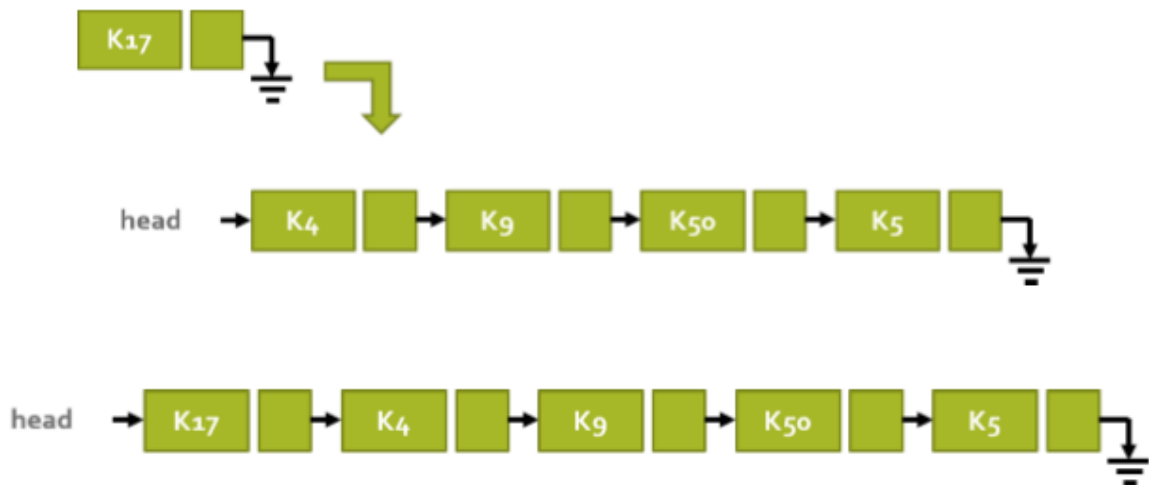
Funciona buscando el primer elemento del dato que solicitamos dentro de la lista que escogimos, si se logra hallar, se regresa un apuntador al elemento en caso contrario se obtiene un NULL. La lista puede tener de 1 a n elementos, los cuales pueden ser recorridos de manera lineal, através de una referencia, teniendo en cuenta que dicha referencia empieza en el primer elemento de la lista (HEAD).

*NOTA: Si la lista se encuentra vacía (sin elementos) no es posible realizar la función "BUSCAR".



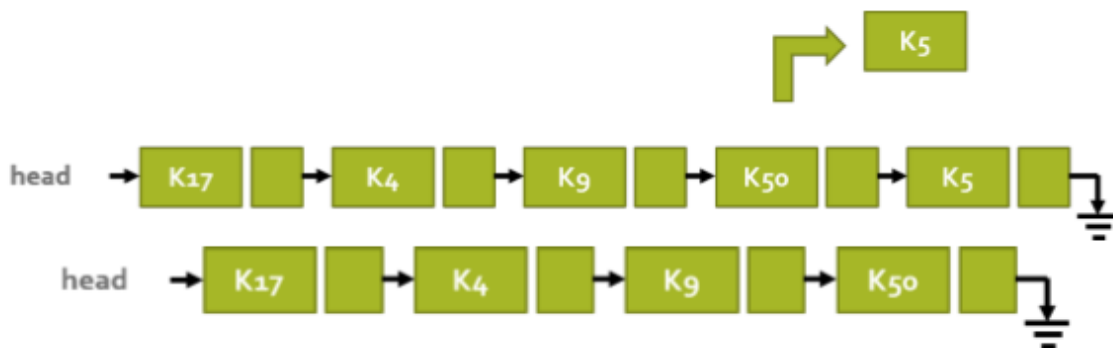
Función - Insertar

Si se tiene un nodo, el cual cumpla con las características de la lista simple, se puede insertar dicho nodo a la lista, ya sea vacía o con elementos. Si se presenta el primer caso, la referencia de dicha lista estará en el nodo que se ingreso recientemente.



Función - Borrar

La función “ELIMINAR” solo es posible en listas con elementos, ya que primero se debe de BUSCAR el nodo que se desea eliminar con el fin, de conocer la ubicación de este. Cuando se haya encontrado el nodo (suponiendo que el nodo existe) se procede a realizar cambios en las referencias, de tal forma que; el antecesor del nodo a eliminar apunte al sucesor de este.



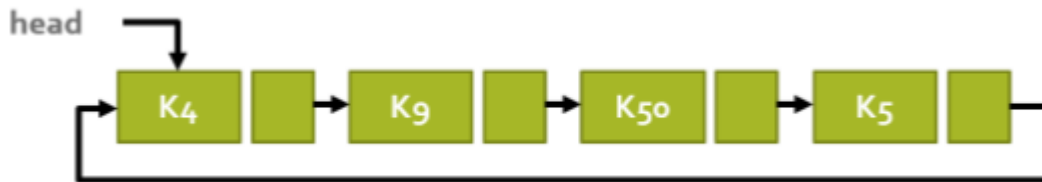
Lista circular

La lista circular es una modificación de la lista simple, donde el apuntador del último elemento (TAIL) apunta al primero (HEAD) de la lista.



Función - Buscar

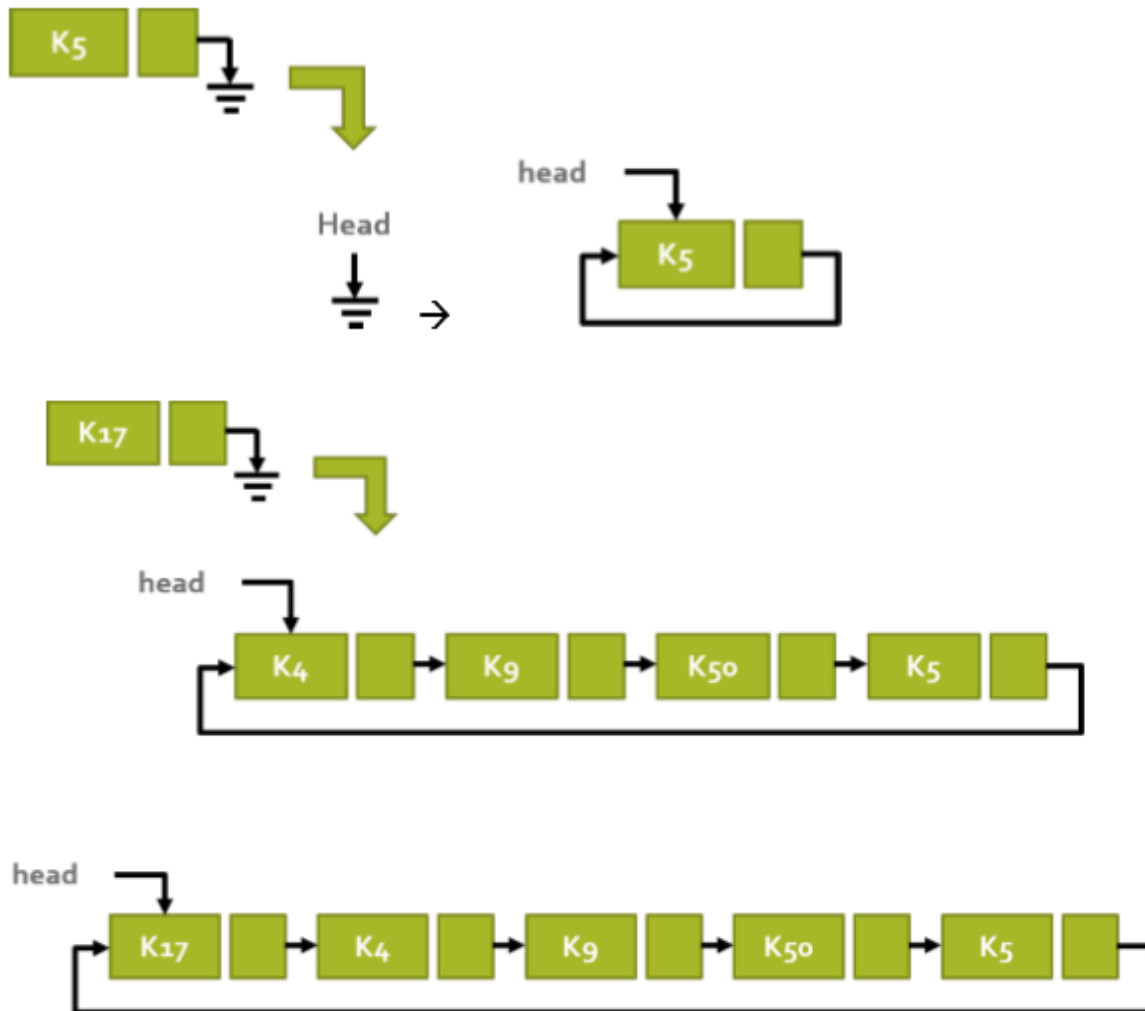
Funciona de manera similar que la lista simple, se busca un parámetro dado, el cual se verifica si existe en la lista. En caso de que se encuentre, se regresa la dirección, en caso contrario, se dice que el nodo no existe en la lista.



Para buscar en una lista, se necesita que esta contenga elementos. Si tiene elementos, puede ser de 1 hasta n, en tal caso la lista se puede recorrer de uno en uno, con base en la referencia. Pero hay tener en cuenta el número de elementos, ya que como la lista es circular; la búsqueda se puede volver infinita.

Función - Insertar

Dicha función es similar a la de la lista simple, con los mismos casos, por lo que no es necesario repetir la teoría, solo se presentará la descripción gráfica de esta función.



Función - Borrar

Para eliminar un nodo de la lista, primero se necesita buscar la referencia de este, para proseguir con la eliminación de este, consiguiendo que el antecesor del nodo a eliminar apunte al siguiente elemento del mismo.

Es necesario que la lista contenga elementos si se desea eliminar algún nodo en específico.





Desarrollo

Aplicaciones

Listas simples

- 1.- La página de la Facultad usa listas simples, ya que uno puede recorrer de una en una el total de sus publicaciones y en algún momento se llegará al final de las publicaciones.
- 2.- Las notificaciones del celular o de cualquier dispositivo electrónico están programadas con listas simples, porque ves una por una las notificaciones, hasta llegar el momento en el que no se encuentran más.
- 3.- En los scanner de los automóviles, cuando se realiza un chequeo al auto, la información que el scanner arroja, esta en forma de lista, ya que una revisión va en seguida de la otra y tiene inicio y fin.

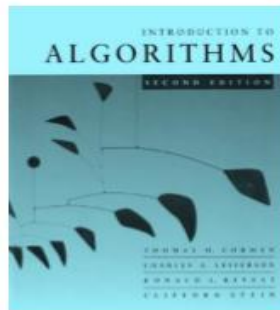
Listas circulares

- 1.- Una lista de canciones se puede reproducir de manera ordena 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.
- 2.- Los trenes del metro usan listas circulares, cuando llegan a la terminal de la línea, se van al fondo de esta, se dan una vuelta y cambian de carril, ahora en el cambio de carril se consiguió un cambio de dirección, que viéndolo de esa forma ahora, está iniciando la ruta.
- 3.- Cuando se programan fuentes de diversión a partir de agua, el agua que sale como espectáculo de la fuente, al caer al suelo se acumula en un depósito, el cual suministrara más tarde agua para la fuente, lo cual continuará de manera sucesiva hasta que la fuente se apague, por lo tanto, es una lista circular, porque el agua acumulada es la misma que saldrá en la fuente.

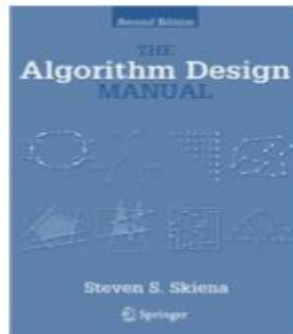
Conclusiones

Las listas son unas estructuras de gran utilidad, aunque le veo mucha similitud con las colas circulares, con estructuras como estas, se puede crear un programa de manera más eficiente y de manera dinámica. La forma dinámica es de gran ayuda para manipular datos de los cuales no se sabe su tamaño.

Bibliografías



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



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

@die-fi-unam [Figura 1]. Consulta: Enero de 2016. Disponible en: <https://twitter.com/die-fi-unam>

Online Media Technologies Ltd. AVS Media Player [Figura 2]. Consulta: Enero de 2016. Disponible en: <https://twitter.com/die-fi-unam>