



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):

8

Integrante(s):

Vega Gutierrez Ricardo Daniel

*No. de Equipo de
cómputo empleado:*

9

No. de Lista o Brigada:

40

Semestre:

2020-2

Fecha de entrega:

24 de marzo del 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo

Se conocerá el funcionamiento y la definición de una lista doblemente ligada y de una lista doblemente ligada circular; con el fin de aplicarlas a situaciones de la vida cotidiana y a usarlas en programas.

Introducción

Las listas son estructuras lineales, porque sus elementos se presentan de manera continua; uno detrás de otro y a su vez son dinámicas, porque su tamaño puede ser modificado durante la ejecución del código, las listas reservan memoria en el stack de la pila.

Lista doblemente ligada

Una lista doblemente ligada, presenta como, unidad básica al nodo. En conjunto los nodos están formados de manera lineal para generar una lista, “doblemente ligada”, quiere decir: que presenta dos referencias una al sucesor y otra al predecesor.



*Nota: Sea un elemento x de la lista

- 1.- El elemento x es el primero de la lista, cuando su predecesor apunta a NULL, lo que significa que la lista no cuenta con más elementos.
- 2.- El elemento x es el último de la lista, si el sucesor de dicho elemento apunta a NULL, en otras palabras, x es el último elemento (TAIL).

Las listas presentan 3 operaciones: buscar, insertar y borrar. Para cada una de las operaciones es necesario presentar dos casos:

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

La operación **buscar** funciona de la siguiente manera; se propone un elemento a buscar, el programa busca relaciones o similitudes entre el elemento propuesto y alguno de la lista, esto lo puede hacer, de inicio a fin o viceversa, siempre recorriendo de uno en uno los nodos.

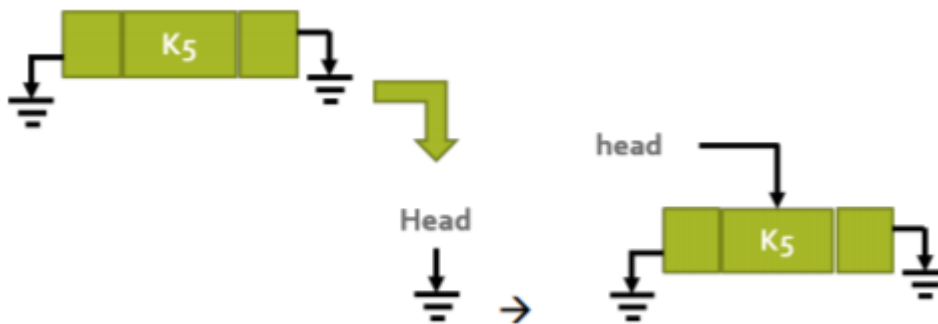


En una lista vacía no se puede buscar, porque no presenta elementos.

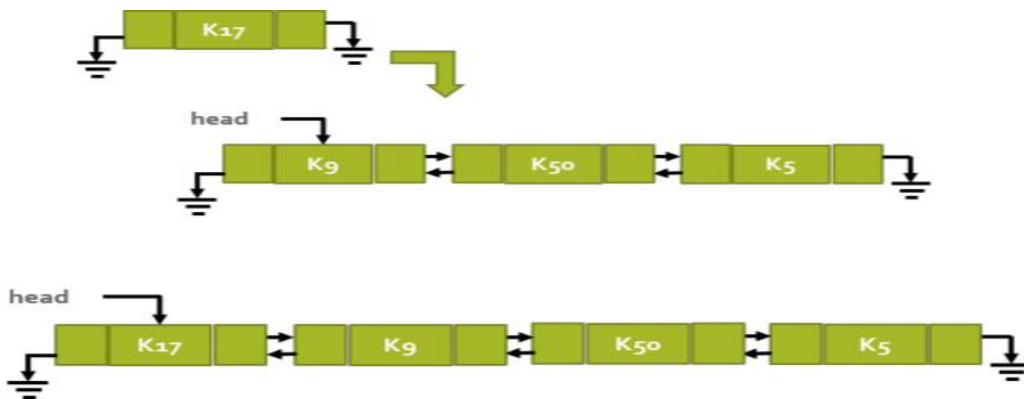


Operación **insertar**, se presenta en los dos casos bases.

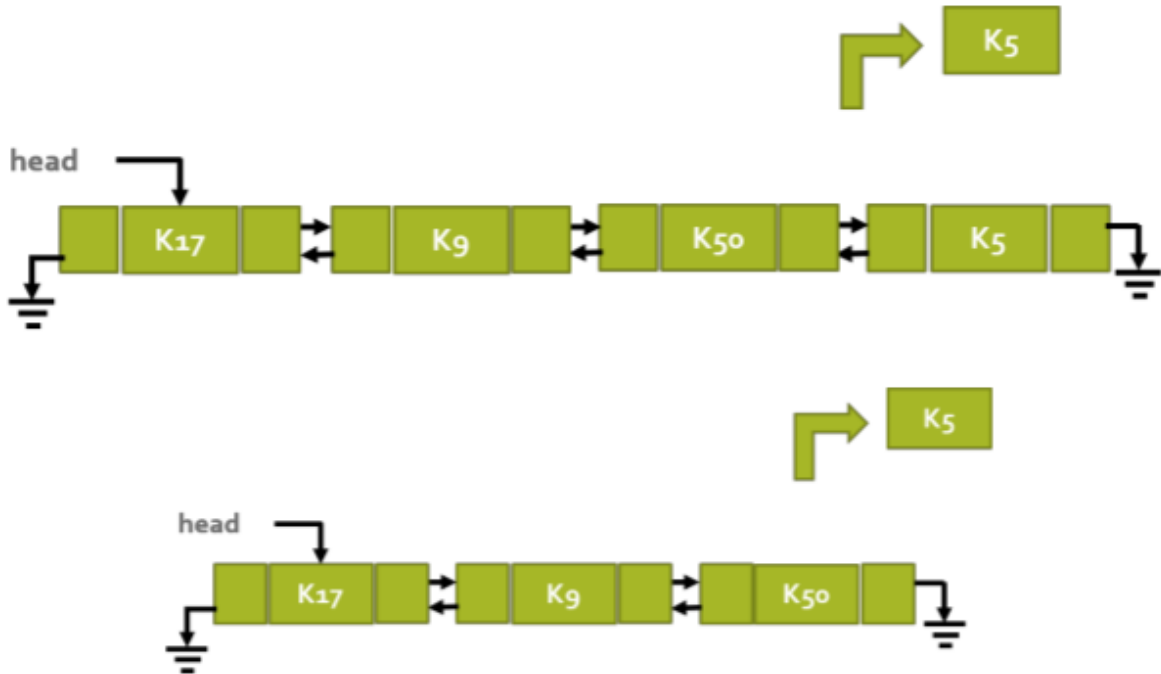
Cuando la lista se encuentra vacía, el elemento agregado tiene las dos referencias apuntando hacia si mismo, las cuales son: (HEAD Y TAIL).



Si la lista cuenta con elementos, el elemento que se desea ingresar puede estar al final o inicio, según lo prefiera, si se agrega al inicio; el elemento tendrá la referencia (HEAD), en caso contrario; hacia el elemento estará apuntando (TAIL).



La operación **borrar**, solo se presenta en listas con elementos, porque no es posible eliminar algún elemento de una lista vacía, ya que no cuenta con alguno. En una lista con elementos, se puede eliminar el primer elemento o el último, según convenga.



Lista doblemente ligada circular

Es circular porque el último elemento (TAIL), apunta al primer elemento de la lista (HEAD), en vez de hacerlo a NULL.



Presenta las mismas operaciones.

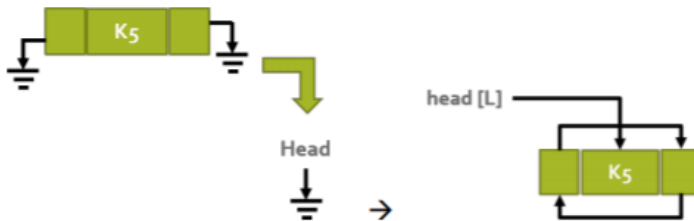
Operación **buscar**, se pone el elemento que se desea buscar y también decir, si se quiere hacer del principio al final o viceversa. Tomando el primer caso, buscar un elemento empezando desde el primer elemento de la lista; se debe tener las dos referencias HEAD y TAIL, cada que se recorra hacia otro elemento se debe comprobar si hay el elemento buscando es el mismo al que estamos apuntando en dicho instante, la búsqueda debe parar al llegar a TAIL.



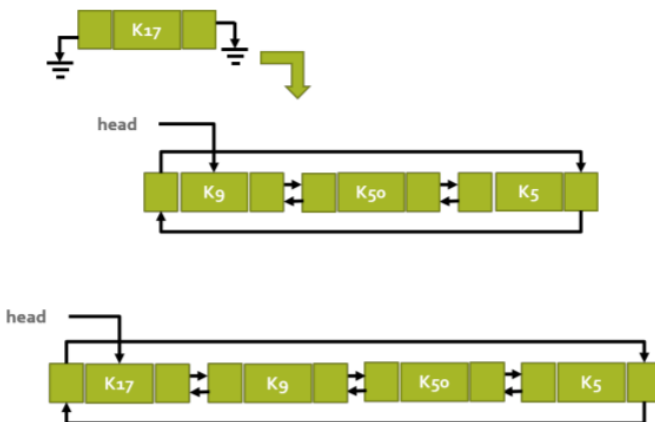
En listas vacías no es posible buscar elementos.



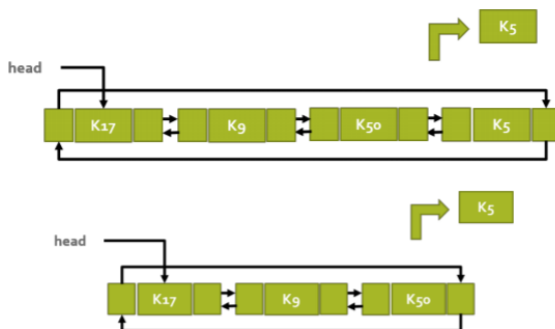
La operación **insertar**: si la lista es vacía, el elemento ingresado es el único, por lo tanto, HEAD y TAIL apuntan al mismo elemento, o sea el nuevo elemento.



Cuando la lista no es vacía, se puede agregar elementos al inicio o final de esta, teniendo en cuenta que es una lista circular.



Por último se encuentra la operación **BORRAR**, si se decide borrar el primer elemento; HEAD se recorre al sucesor del elemento borrado y ahora el último elemento apunta al mismo que HEAD. Si se borra el último elemento, la referencia TAIL apunta hacia el predecesor del elemento borrado y a su vez, éste elemento apunta a HEAD. En listas sin elementos no es posible llevar a cabo la función borrar.



Desarrollo

Aplicaciones de:

1.- Lista doble

- Cuando checas tu correo lo haces desde la parte superior, vas bajando poco a poco, hasta llegar al último correo, al llegar al final de los correos, no es posible seguir avanzando, porque se llegó al TAIL de la lista
- Cuando se ocupa la operación suma con el símbolo sigma (Σ), se suma un número determinado de elementos, dicha suma va desde el inicio y suma de uno en uno hasta llegar al índice final de la suma. Una vez que se suma el último elemento no es posible regresar al primer elemento que se sumó.
- En una pista de carreras, en la competencia de 100 m planos, al llegar a la meta, no es posible llegar al comienzo de la carrera.

2.- Lista doble circular

- En una play list de Youtube cada video tiene la opción de regresar al anterior o reproducir el siguiente.
- Las escaleras electrónicas tienen HEAD parte donde ingresas a la escalera y TAIL parte donde sales de la escalera, funcionan de manera infinita, porque la escalera que esta en HEAD debe llegar a TAIL y al llegar a este lugar, ahora debe de regresar a su posición original y así sucesivamente, sin detenerse a menos que un factor externo intervenga.
- Cuando la NASA envía determinada onda a un lugar, espera que regrese dicha onda, según las características que tenga la onda recibida, será las hipótesis que realicen.

Conclusiones

Las listas dobles y dobles circulares son muy similares a las listas que en la práctica anterior se vieron, solo que con un funcionamiento distinto. Por lo que, uno decide que tipo de listas usar para dar solución a un problema. El trabajar con listas dobles amplía mi conocimiento, para poder tener más información a la hora de resolver un problema.

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. Google (2016). Gmail [Figura 1 y figura 2]. Consulta: Enero de 2016. Disponible en: <https://www.mail.google.com> Youtube (2016). Mariachi [Figura 3]. Consulta: Enero de 2016. Disponible en: <https://www.youtube.com/watch?v=qKEm19IMjuQ&list=PL84EC9ACDAF6B300C> Youtube (2016). Mariachi [Figura 4]. Consulta: Enero de 2016. Disponible en: <https://www.youtube.com/watch?v=13dnkytPEZQ&index=100&list=PL84EC9ACDAF6B300C>