SESIÓN **/07**

LISTAS CIRCULARES Y DOBLES

1. Listas enlazadas circulares
2. Listas enlazadas dobles

**INTRODUCCIÓN**

* La clase anterior tratamos en detalle las operaciones sobre una Lista enlazada simple. Describimos los algoritmos de búsqueda, modificación, eliminación y ordenamiento.
* En la presente sesión conoceremos los otros dos tipos de Listas enlazadas: Las listas circulares y las Listas dobles. Revisaremos sus principales características, su representación gráfica y las operaciones que se pueden realizar con estas estructuras.
* Culminaremos comparando los algoritmos de los tres tipos de Listas enlazadas.

1. **LISTAS ENLAZADAS CIRCULARES**
   1. **Características**

* Es similar a las Listas enlazadas simples, cada nodo tiene dos partes: un campo de datos y un campo de enlace.
* El campo de enlace del último nodo no apunta a un valor nulo, apunta al primer elemento de la Lista.
* Se utiliza una variable llamada último, la cual guarda la dirección de memoria del último elemento de la Lista.

dato

dato

dato

último

dato

* 1. **Representación gráfica**

La variable último apunta al último elemento de la Lista circular.

* 1. **Operaciones**

En una Lista enlazada circular se pueden realizar las mismas operaciones que se realizaron sobre una Lista enlazada simple.

* + 1. Algoritmo para Recorrido

1. Ubicarse en el primer nodo de la Lista.
2. Repetir los pasos 3 y 4 hasta llegar al último nodo.
3. Mostrar los datos contenidos en el nodo.
4. Pasar al siguiente nodo.
5. Mostrar los datos del último nodo de la Lista.
   * 1. Algoritmo para Búsqueda
6. Ubicarse en el primer nodo de la Lista.
7. Repetir los pasos 3 y 4 hasta encontrar el dato ó hasta llegar al último nodo.
8. Comparar el dato buscado con el dato contenido en el nodo.
9. Pasar al siguiente nodo de la Lista.
10. Comparar el dato buscado con el dato contenido en el último nodo.
    * 1. Algoritmo para Inserción

Antes de insertar un nodo, se debe verificar si la Lista está o no vacía.

Si la Lista está vacía, se puede utilizar el siguente algoritmo:

1. Asignar memoria para el nuevo nodo.
2. Asignar el valor para el campo de datos.
3. Hacer que el campo de enlace apunte al nuevo nodo.
4. Hacer que la variable último apunte al nuevo nodo.

Si la Lista no está vacía, se puede insertar un nodo en las siguientes ubicaciones:

* Al inicio de la Lista.
* Al final de la Lista.
* Entre dos elementos.

Estos algoritmos se describieron en las Listas enlazadas simples.

* + 1. Algoritmo para Eliminación

Antes de implementar esta operación, es necesario verificar:

* Si la Lista está vacía, se deberá mostrar un mensaje.
* Si la Lista no está vacía, se buscará el nodo a eliminar.
* Si el nodo no está en la Lista, se deberá mostrar un mensaje.

Si el nodo se encuentra en la Lista, puede estar en una de las siguientes ubicaciones:

* Al inicio de la Lista.
* Al final de la Lista.
* Entre dos elementos.

Estos algoritmos se describieron en las Listas enlazadas simples.

1. **LISTAS ENLAZADAS DOBLES**
   1. **Características**

* Se les conoce también como Listas doblemente enlazadas.
* Cada elemento contiene la dirección de memoria del siguiente elemento de la Lista así como la dirección de memoria del elemento anterior.
* Tienen la flexibilidad de poder recorrerse en ambas direcciones: del inicio al final ó viceversa.
  1. **Estructura de un nodo**

La estructura de un nodo consta de 3 campos: ante, info y sgte.

ante

sgte

info

En el campo info se coloca el dato que se desea almacenar en la Lista enlazada.

En el campo sgte se coloca la dirección de memoria del siguiente elemento de la Lista.

En el campo ante se coloca la dirección de memoria del elemento anterior de la Lista.

* 1. **Representación gráfica**

Los nodos se grafican en el mismo orden en el cual se fueron adicionando a la Lista enlazada.

dato

dato

dato

dato

null

null

último

inicio

Se utiliza una variable llamada inicio, la cual contiene la dirección de memoria donde está almacenado el primer elemento. Además, se utiliza una variable llamada último, la cual contiene la dirección de memoria del último elemento de la Lista.

* 1. **Operaciones**

En una Lista enlazada doble se pueden realizar las mismas operaciones que se realizaron sobre una Lista enlazada simple.

* + 1. Algoritmo para Recorrido

1. Ubicarse en el primer nodo de la Lista.

2. Repetir los pasos 3 y 4 hasta llegar al final de la Lista.

3. Mostrar los datos contenidos en el nodo.

4. Pasar al siguiente nodo.

* + 1. Algoritmo para Búsqueda

1. Ubicarse en el primer nodo de la Lista.

2. Repetir los pasos 3 y 4 hasta encontrar el dato o hasta llegar al final de la Lista.

3. Comparar el dato buscado con el dato contenido en el nodo.

4. Pasar al siguiente nodo.

* + 1. Algoritmo para Inserción

Antes de insertar un nodo, se debe verificar si la Lista está o no vacía.

Si la Lista está vacía, se puede utilizar el siguente algoritmo:

1. Asignar memoria para el nuevo nodo.
2. Asignar el valor para el campo de datos.
3. Hacer que el campo de enlace sgte apunte a null.
4. Hacer que el campo de enlace ante apunte a null
5. Hacer que la variable inicio apunte al nuevo nodo.

Si la Lista no está vacía, se puede insertar un nodo en las siguientes ubicaciones:

* Al inicio de la Lista.
* Al final de la Lista.
* Entre dos elementos.

Estos algoritmos se describieron en las Listas enlazadas simples.

* + 1. Algortimo para Eliminación

Antes de implementar esta operación, es necesario verificar:

* Si la Lista está vacía, se deberá mostrar un mensaje.
* Si la Lista no está vacía, se buscará el nodo a eliminar.
* Si el nodo no está en la Lista, se deberá mostrar un mensaje.

Si el nodo se encuentra en la Lista, puede estar en una de las siguientes ubicaciones:

* Al inicio de la Lista.
* Al final de la Lista.
* Entre dos elementos.

Estos algoritmos se describieron en las Listas enlazadas simples.

**CONCLUSIÓN**

La diferencia entre una Lista simple y una Lista circular, está en el campo de enlace del último elemento. Por otro lado, las Listas dobles se diferencian de los otros tipos de Listas por la estructura de su nodo. Por último, los algoritmos de mayor complejidad son los que se aplican en una Lista doble, mientras que los algoritmos de menor complejidad son los que se aplican en una Lista simple.

**REFERENCIAS BIBLIOGRÁFICAS**

* Cairo, O.; Guardati, S. (2008). Estructuras de datos. 3ra. Edición. México D.F., Mexico: McGraw Hill.
* Instituto NIIT (2011). Data Structures and Algorithms. Student guide.