Republica Bolivariana de Venezuela

Universidad Rafael Urdaneta

Facultad de Ingeniería

Estructuras Dinámicas de Datos

Sección: A

Profesor: Eli Mora

Informa del contenido de la exposición de Listas y Colas

Nombre: Jesus Ortega

C.I.: 29.891.702

Fecha de entrega: 09/07/2022

1. Colas
   1. Estructuras dinámicas “Colas”
   2. Definición de una cola: una cola es una estructura de datos en la que el primer elemento en entrar es el primero en salir. Las colas funcionan como una sucesión de nodos, en el que cada nodo apunta a otro nodo, de forma que en una cola los datos siempre van a entrar por el mismo extremo y saldrán por el otro.
   3. Estructura de una cola: la estructura de una cola,es formada por dos nodos uno por cada extremo de la misma, estos dos nodos principales serán los usados para poder guardar y acceder a los elementos de la cola cuando lo deseemos.
   4. Nodos de una cola: Los nodos que se encuentran en una cola deben apuntar siempre a otro nodo que ira después de este, asimismo este guardara la información que se deseen reservar en la cola. Para crear un nodo se debe reservar el espacio de memoria que requiera la creación de su estructura, así como asignarle la información que se quiera guardar y asegurarse que este no apunte a otro nodo hasta que se introduzca en la cola, es decir que, al crearse un nodo, su nodo siguiente será NULL.
   5. Creación de una cola: Para crear una cola debemos reservar el espacio que esta ocupara, y después hacer que sus extremos sean iguales a NULL, debido a que esta se encuentra vacía.
2. Como se insertan elementos en una cola.

Para agregar un nodo a una cola debemos crear el nodo que deseamos agregar con la información requerida y hacer que el ultimo elemento de la cola apunte a este nuevo nodo, de forma que ahora el nuevo ultimo elemento de la cola sea el nodo introducido ya que este no apuntara a ningún otro nodo.

1. Eliminar elementos en una cola.  
   Para eliminar elementos de una cola debemos crear un puntero auxiliar que nos permita acceder al primer nodo de la cola, el cual será el primero ya que en una cola, por definición siempre se eliminara el primer elemento al ser este el que lleva ultimo en ser introducido. Una vez guardado el primer nodo, hacemos que el segundo nodo de la cola pase a ser el primero, y finalmente liberamos el nodo eliminado de la memoria.
2. Listas
   1. Estructuras dinámicas “Listas”
   2. Definición: una lista es una estructura de datos que represente una colección de elementos ordenados, una lista puede ser implementada de diversas formas, una de las implementaciones mas comunes son las listas enlazadas, en la que se enlazan nodos de forma sucesiva. Los nodos en una lista tendrán una estructura igual o similar a los de una cola y se crearán de la misma forma.
   3. Estructura de una lista: la estructura de una lista estará compuesta principalmente de una cabeza, el cual apuntará siempre al primer elemento de la lista, esta cabeza será utilizada para acceder al resto de los nodos de la lista. También se pueden colocar otros datos que faciliten el proceso que queramos hacer, como puede ser una variable que guarde la longitud de la lista.
   4. Creación de una lista : para crear una lista primero se debe reservar la memoria de la misma, posteriormente se le asignan los valores a los datos, como pueden ser la cabeza o la longitud, que serán igual a NULL y 0 respectivamente debido a que aun no se han agregado nodos.
   5. Como se insertan elementos en una lista.

A diferencia de la cola en una lista pueden insertarse nodos en cualquier posición de la propia lista, aunque el procedimiento puede cambiar dependiendo si se desea introducir al comienzo, al final o en algún punto intermedio. A continuación, se explicarán algunas como introducir datos dependiendo del lugar en el que queramos introducir:

* Introducir un nodo al comienzo de una lista: para este caso luego de crear el nuevo nodo, haremos que este tenga como nodo siguiente la cabeza de la lista, de modo que ahora la nueva cabeza de la lista es el nodo recién creado.
* Introducir un nodo en cualquier punto de la lista: se usa un puntero que nos permita recorrer la lista desde la cabeza hasta el lugar en el que queramos agregar el nodo. Posteriormente hacemos que el nuevo nodo apunte al mismo nodo al que apunta el puntero auxiliar, y después que el puntero auxiliar apunte al nuevo nodo, de esta forma el nodo estará justo después de la posición del puntero auxiliar y se habrá agregado a la lista correctamente.
* Introducir un nodo al final de la lista: se hará de forma similar al realizado para introducir en cualquier parte de la lista, con la diferencia de que como el puntero auxiliar deberá apuntar al ultimo nodo de la lista, el nuevo elemento tendrá un valor siguiente igual a NULL.  
  1. Eliminar elementos en una lista.  
     El espacio de memoria usado por un nodo puede ser liberado usando la función free(). Sin embargo, antes de liberar el espacio del nodo, debemos hacer que el nodo que estaba antes del que deseamos eliminar apunte ahora al siguiente del nodo eliminado.  
     Es importante destacar que se debe crear un puntero que guarde el nodo que queremos eliminar, ya que no podremos acceder al nodo directamente una vez alteradas las “flechas” de los nodos. Por lo tanto, usaremos ese puntero auxiliar para liberar la memoria del nodo, de forma similar a la utilizada por las colas.

De forma similar al momento de introducir elementos, también habrá que tomar en cuenta que el primer elemento es accedido directamente desde la lista y que este no tiene un nodo que le preceda, al igual que el ultimo nodo apunta a un valor NULL