

10 ESTRUCTURAS DE DATOS ELEMENTALES

10.1 Pila y Colas

- Los elementos que se remueven de la lista son específicos;

Pilas

- En el Pila el elemento a borrar es el último elemento que ingresó a la lista implementa un last-in y un first-out o LIFO
- La operación de insertar es llamado PUSH
- La operación de borrado es llamado POP
- Tiene un identificador TOP
- El orden como podemos acceder a los datos es siempre la inversa de como los apilamos
- Si intentamos acceder a un dato de una pila vacía, se le nombra stack underflows, que es normalmente un error
- Si excede la mayor cantidad de datos que puede tener nuestra fila se le nombra the stack overflows

Colas

- El elemento a borrar es el elemento que más tiempo ha estado en la lista
- Implementa un first-in y un first-out o FIFO
- La operación de insertar es llamado ENQUEUE
- La operación de borrado es llamado DEQUEUE
- Inspirado en una lista de clientes esperando pagar
- La cola tiene una head y una tail. Cuando un nuevo dato entra a la lista, toma el lugar de la cola(tail).El elemento borrrable si es el elemento que más tiempo a estado en la cola(head)
- Si head y tail son nulos la lista está vacía (underflows)
- Si $head = tail + 1$, la lista está llena, si se ingresa un nuevo dato se excede espacio (overflows)

10.2 Listas Enlazadas

Listas enlazadas

- Los objetos son ordenados en un orden lineal
- A diferencia de un arreglo, el cual su orden es determinado por su el índice del arreglo. En una lista enlazada es determinado por un apuntador en los objetos
- Arreglo dinámico
- Una lista de doble enlace, es un objeto con un atributo y dos punteros (anterior y siguiente)
- Si un elemento en su parte de anterior es NULL, significa que es el inicio de la cadena
- Si un elemento en su parte de siguiente es NULL, es el final de la lista
- Si anterior como siguiente es NULL, entonces son NULL hay solo un elemento, pero su el dato es NULL la lista está vacía
- *Ejercicios:*
 - Buscar un número en una lista: Si no se encuentra regresa NULL
 - Insertar un nuevo nodo: inserta un nodo enfrente de la lista
 - Borra un elemento de la lista: Es necesario un puntero

Centinela: Nos permite crear un nuevo nodo que hace que una lista de doble enlace se interpreta como una lista circular uniendo el inicio y final de la lista.

10.3 Implementación de punteros y objetos

Podemos representar una serie de objetos que cuenta con las mismas características por medio de una matriz.

Representación multi-arreglo de objetos

- Se presenta un arreglo donde se guardará cualquier dato
- Los punteros son relaciones entre los arreglos que residen el anterior y siguiente
- Dado un índice (x), el arreglo entra con $\text{dato}[x]$, $\text{anterior}[x]$ y $\text{siguiente}[x]$

Representación unidimensional de objetos

- La memoria de una computadora es un arreglo unidimensional con direcciones enteras de 0 hasta $N-1$. Un apuntador nos sirve para indicar la primera dirección donde el objeto se localiza, aunque también podemos localizar otras direcciones, agregando otro acople al puntero.
- El representación unidimensional nos permite hacer insertar objetos de diferentes longitudes

Asignación y liberación de objetos

- Es importante llevar una administración de los objetos que no estén relacionados a la lista
- El *recolector de basura* es responsable de determinar qué objetos se desvían.
- Los objetos libres estarán en una lista enlazada simple, la cual se llamará free list. La cabecera de la lista siempre se guarda en una variable global free.
- Cuando la lista no sea NULL, la lista free puede estar enlazada
- La lista free actúa como un stack.

10.4 Representando de árboles enraizados

Arbol Binario

- Si usamos los atributos p , left y right . Si se mete un nuevo nodo X y $x.\text{left}=\text{NULL}$ y $x.\text{right}=\text{NULL}$ entonces $T.\text{root}=x$; Si la $T.\text{root}=\text{NULL}$ entonces el árbol está vacío.

Árboles enraizados con ramificación ilimitada

- Teniendo un número arbitrario de niños podemos hacer un esquema donde cada nodo del árbol tenga un elemento p (que representa el dato a guardar) con dos punteros.
- $x.\text{left-child}$: apunta a el niño de la izquierda del nodo x
- $x.\text{right-siblings}$: apunta a los hermanos de x inmediatamente a su derecha