



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA y ELÉCTRICA

UNIDAD CULHUACAN

INGENIERÍA EN COMPUTACIÓN

ASIGNATURA: ESTRUCTURA DE DATOS

Tarea 1

Suarez Vega Edgar Alan

Profesora: Clarissa Janeth González Acatitla

Semestre 2-2021

Febrero - Junio 2021

Actividad 1 :

- **Que son las estructuras de datos**

Las estructuras de datos son una forma de organizar los datos en la computadora mediando uso de un lenguaje de programación, de tal manera que nos permita realizar unas operaciones con ellas de forma muy eficiente. Es decir, igual que un array introducimos un dato y eso es prácticamente inmediato, no siempre lo es, según qué estructuras de datos y qué operaciones.

Depende que algoritmo queramos ejecutar, habrá veces que sea mejor utilizar una estructura de datos u otra estructura que nos permita más velocidad.

Estructuras de datos mas conocidas :

- Arrays
- Montículos binarios
- Pilas
- Colas

- **Aplicaciones de las estructuras de datos y ejemplos**

Pueden ser desde programas que muestren listas de alumnos y manejen el uso de pocos datos, hasta programas mas complejos que usen muchas cantidades de datos como sistemas de control de usuarios. Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas

Arrays

La estructura de datos más simple es el array lineal (o unidimensional). Un array lineal es una lista de números finitos de datos similares, referenciados por medio de un conjunto de n números consecutivos, normalmente $1, 2, 3, \dots, n$.

Pila

Una pila, también denominada sistema último-dentro primero-fuera (LIFO), es una lista lineal en la cual las inserciones y extracciones tienen lugar sólo por un extremo llamado cúspide.

Cola

Una cola, también denominada sistema primero-dentro primero-fuera (FIFO), es una lista lineal en la cual las extracciones se realizan siempre por un extremo llamado frente y las inserciones por el extremo contrario llamado final de la lista.

Grafos

Los datos contienen, en algunos casos, relaciones entre ellos que no es necesariamente jerárquica. Por ejemplo, supongamos que unas líneas aéreas realizan vuelos sólo entre ciudades conectadas por líneas. La estructura de datos que refleja esta relación recibe el nombre de grafo

- **Organización elemental de los datos.**

Los datos naturalmente se organizan de acuerdo a sus características, si son palabras o letras sería alfabéticamente o si son números podría ser de mayor a menor y viceversa o incluso un sin fin de características que compartan dichos números a ordenar, otras formas de organizar datos es de manera cronológica, que depende de en que momento sean ingresados o jerárquica , conforme a que dato sea mas relevante o importante.

- **Descripción de la terminología, dato, registro, grupo de elementos, elemento simple, entidad, atributo, rango de valores, campo, arreglo, listas, colas, pilas y árboles.**

Dato : Un dato es la representación de una variable que puede ser cuantitativa o cualitativa que indica un valor que se le asigna a las cosas y se representa a través de una secuencia de símbolos, números o letras. Existen diferentes tipos de datos como int, float , etc.

Registro : Un registro en programación, es un tipo de dato estructurado formado por la unión de varios elementos bajo una misma estructura. Estos elementos pueden ser, o bien datos elementales (entero, real, carácter,...), o bien otras estructuras de datos. A cada uno de esos elementos se le llama campo. (STRUCT).

Grupo de elementos : Son elementos del mismo tipo ordenados o juntos que comparten ciertas características o atributos que los hacen relacionarse, Con una estructura de datos ideal se comprueba si un elemento se encuentra en el conjunto.

Elemento simple : Los elementos simples solamente pueden contener texto (caracteres). Dicho de otro modo, los elementos simples no pueden contener a otro u otros elementos (hijos), ni tampoco pueden tener atributos.

Entidad : Una entidad es un objeto concreto o abstracto que presenta interés para el sistema y sobre el que se recoge información la cual va a ser representada en un sistema de base de datos. La mayoría de las entidades modelan objetos o eventos del mundo real, por ejemplo, clientes, productos o llamadas de pedidos.

Atributo : En informática, un atributo es una especificación que define una propiedad de un objeto, elemento o archivo. También puede referirse o establecer el valor específico para una instancia determinada de los mismos.

Rango de valores : son todos los posibles valores que puede tener un elemento, objetos, etc. que le permite el software o el programa.

Campo : Un campo es el nombre de la unidad de información. Cada entrada en una base de datos puede tener múltiples campos de diversos tipos. Por ejemplo, un campo de texto llamado 'color favorito', que permite escribirlo, o un menú llamado 'población' que permita escoger de un listado de poblaciones posibles.

Arreglo : es un conjunto de datos o una estructura de datos homogéneos que se encuentran ubicados en forma consecutiva en la memoria RAM (sirve para almacenar datos en forma temporal).

Lista : La Lista es una estructura de datos muy importante en los lenguajes de programación donde: representa una colección de elementos ordenados. puede contener elementos repetidos. cada elemento de la lista tiene un índice que lo ubica dentro de la misma.

Lista es una secuencia conectada de nodes, cada uno de los cuales contiene algún dato. Hay un nodo al comienzo llamado la cabeza o frente (head o front). Hay un nodo de término llamado la cola o atrás (tail o back).

Colas : Una cola es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pull por el otro. También se le llama estructura FIFO, debido a que el primer elemento en entrar será también el primero en salir.

Pilas : Una pila (stack en inglés) es una lista ordenada o estructura de datos que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO (del inglés Last In, First Out, «último en entrar, primero en salir») . Esta estructura se aplica en multitud de supuestos en el área de informática debido a su simplicidad y capacidad de dar respuesta a numerosos procesos.

Tipos de colas :

- Colas circulares (anillos): en las que el último elemento y el primero están unidos.
- Colas de prioridad: En ellas, los elementos se atienden en el orden indicado por una prioridad asociada a cada uno. Si varios elementos tienen la misma prioridad, se atenderán de modo convencional según la posición que ocupen. Hay 2 formas de implementación:
- Bicolos: son colas en donde los nodos se pueden añadir y quitar por ambos extremos; se les llama DEQUE (Double Ended QUEUE). Para representar las bicolas lo podemos hacer con un array circular con Inicio y Fin que apunten a cada uno de los extremos. Hay variantes:
- Bicolos de entrada restringida: Son aquellas donde la inserción sólo se hace por el final, aunque podemos eliminar al inicio ó al final.
- Bicolos de salida restringida: Son aquellas donde sólo se elimina por el final, aunque se puede insertar al inicio y al final.

Arboles : Los árboles son estructuras de datos muy similares a las listas doblemente enlazadas, en el sentido que tienen punteros que apuntan a otros elementos, pero no tienen una estructura lógica de tipo lineal o secuencial como aquellas, sino ramificada. Tienen aspecto de árbol, de ahí su nombre.

Tipos:

Binarios : un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos. Si algún hijo tiene como referencia a null, es decir que no almacena ningún dato, entonces este es llamado un nodo externo.

AVL : Los árboles AVL están siempre equilibrados de tal modo que para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha o viceversa. ... Los árboles AVL más profundos son los árboles de Fibonacci.

Rojo – Negro : Un árbol rojo-negro es un árbol binario de búsqueda en el que cada nodo tiene un atributo de color cuyo valor es rojo o negro. En adelante, se dice que un nodo es rojo o negro haciendo referencia a dicho atributo.

Además de los requisitos impuestos a los árboles binarios de búsqueda convencionales, se deben satisfacer las siguientes reglas para tener un árbol rojo-negro válido:

Todo nodo es o bien rojo o bien negro.

La raíz es negra.

Todas las hojas (NULL) son negras.

Todo nodo rojo debe tener dos nodos hijos negros.

Cada camino desde un nodo dado a sus hojas descendientes contiene el mismo número de nodos negros..

• Operaciones con estructuras de datos elementales

Una estructura de datos define la organización e interrelación de éstos y un conjunto de operaciones que se pueden realizar sobre ellos. Las operaciones básicas son:

Alta : adicionar un nuevo valor a la estructura.

Baja : borrar un valor de la estructura.

Búsqueda : encontrar un determinado valor en la estructura para realizar una operación con este valor, en forma SECUENCIAL o BINARIO (siempre y cuando los datos estén ordenados)...

Otras operaciones que se pueden realizar son:

Ordenamiento : de los elementos pertenecientes a la estructura.

Apareo : dadas dos estructuras originar una nueva ordenada y que contenga a las apareadas.

Cada estructura ofrece ventajas y desventajas en relación a la simplicidad y eficiencia para la realización de cada operación. De esta forma, la elección de la estructura de datos apropiada para cada problema depende de factores como la frecuencia y el orden en que se realiza cada operación sobre los datos.

Tipos de datos elementales

- Binarios
 - Bit
 - Byte
 - Numéricos
 - Entero
 - Real
 - Coma fija
 - Coma flotante
 - Alfanuméricos
 - Carácter
 - Cadena
 - Booleanos
-
- **Descripción de las operaciones, en estructura de datos, de recorrido, búsqueda, inserción y eliminación.**

Recorrido : el recorrido se refiere al proceso de visitar de una manera sistemática, exactamente una vez, cada nodo en una estructura de datos.

Búsqueda : consiste en examinar, a partir del primer elemento y de uno en uno, hasta encontrar el dato buscado o bien llegar al final de la lista que puede estar almacenada en archivo o arreglo.

Inserción : consiste en anexar un nuevo elemento o un establecer el valor de algún atributo de algún elemento, de una estructura de datos.

Eliminación : consiste en quitar un elemento o el valor de algún atributo de algún elemento, de una estructura de datos.

- **Descripción de las operaciones, en estructuras de datos, ordenación y mezcla.**

Ordenación : La ordenación o clasificación de datos (sort en inglés) es una operación consistente en disponer un conjunto de datos en algún determinado orden con respecto a uno de los campos de los elementos del conjunto. Por ejemplo, cada elemento del conjunto de datos de una guía telefónica tiene un campo nombre, un campo dirección y un campo número de teléfono; • En terminología de ordenación, el elemento por el cual está ordenado un conjunto de datos (o se está buscando) se denomina clave.

Mezcla : El ordenamiento por mezcla es un algoritmo recursivo que divide continuamente una lista por la mitad. Si la lista está vacía o tiene un solo ítem, se ordena por definición (el caso base). Si la lista tiene más de un ítem, dividimos la lista e invocamos recursiva mente un ordenamiento por mezcla para ambas mitades.

- **Notación algorítmica, Diagrama de flujo y pseudocódigo**

Notación Algorítmica : es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades.

Diagrama de flujo : Un diagrama de flujo es una representación gráfica de un proceso. Se trata de representar los pasos que sigue un proceso desde que se inicia hasta que se termina y para ello se utiliza una serie de elementos visuales que te ayuden a dibujar cada paso que sigue un proceso.

pseudocódigo : El pseudocódigo es una forma de escribir los pasos que va a realizar un programa de la forma más cercana al lenguaje de programación que vamos a utilizar posteriormente. Es como un falso lenguaje, pero en nuestro idioma, en el lenguaje humano y en español.

- **Estructuras de control y Recursividad.**

Estructuras de control : Las estructuras de control, son instrucciones que permiten romper la secuencialidad de la ejecución de un programa; esto significa que una estructura de control permite que se realicen unas instrucciones y omitir otras, de acuerdo a la evaluación de una condición.

Existen 2 tipos de estructuras de control:

1. Selectivas
2. Repetitivas

Las primeras permiten seleccionar un camino a ejecutarse entre dos o más opciones "por una sola vez"; y las segundas, permiten ejecutar un conjunto de instrucciones "varias veces" mientras se cumpla una condición.

Recursividad : La recursividad consiste en funciones que se llaman a sí mismas, dentro de si mismas o en dentro de otra, evitando el uso de bucles y otros iteradores. Existen dos tipos de recursividad: Directa: Es la más común, se da cuando una función se llama a sí misma una o varias veces. Indirecta: Se da cuando una función es llamada de manera indirecta, es decir, por medio de otra función.

- **Arreglos unidimensionales.**

Están formados por una colección finita y ordenada de datos del mismo tipo. Los datos que se guarden en los arreglos todos deben ser del mismo tipo. Cada elemento se guarda en un espacio independiente.

- **Acceso a las diferentes localidades de un arreglo de caracteres unidimensionales.**

Se accede a cada elemento individual del array mediante un número entero denominado índice. 0 es el índice o localizador del primer elemento y n-1 es el índice del último elemento, siendo n la dimensión, tamaño o longitud del array.

- **Arreglos bidimensionales, matrices.**

Todos los elementos de la matriz deben ser del mismo tipo (int, float, String etc.) Las filas y columnas comienzan a numerarse a partir de cero, similar a los vectores. Es prácticamente un array pero en 2 dimensiones, por consiguiente tiene que tener 2 tamaños, uno para filas y otro para columnas. Se accede a cada localidades de la matriz mediante 2 índices, uno de fila y otro de columnas, que dan un solo espacio específico, es como usar coordenadas, por ejemplo localizar en una matriz 3x3 el primer elemento esta en la posición 0 filas y columnas, ya que empieza desde el 0, sería (1,1).

Actividad 2.

- **Investigar el funcionamiento de la estructura de datos pila, sus aplicaciones y representación con arreglos unidimensionales.**

Una pila es una lista ordinal o estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO que permite almacenar y recuperar datos. Esta estructura se aplica en multitud de ocasiones en el área de informática debido a su simplicidad y ordenación implícita de la propia estructura.

Para el manejo de los datos se cuenta con dos operaciones básicas: apilar (**push**), que coloca un objeto en la pila, y su operación inversa, retirar (o desapilar, **pop**), que retira el último elemento apilado.

En cada momento sólo se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado . La operación retirar permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al siguiente, que pasa a ser el nuevo TOS.

Por analogía con objetos cotidianos, una operación apilar equivaldría a colocar un plato sobre una pila de platos, y una operación retirar a retirarlo.

Una pila cuenta con 2 operaciones imprescindibles: apilar y desapilar, a las que en las implementaciones modernas de las pilas se suelen añadir más de uso habitual.

Crear: se crea la pila vacía. (constructor)

Tamaño: regresa el numero de elementos de la pila. (size)

Apilar: se añade un elemento a la pila.(push)

Desapilar: se elimina el elemento frontal de la pila.(pop)

Cima: devuelve el elemento que esta en la cima de la pila. (top o peek)

Vacía: devuelve cierto si la pila está vacía o falso en caso contrario (empty).

Una pila puede ser fácilmente implementada utilizando arreglos y manteniendo siempre una referencia a la cima, es decir, un índice que indique la posición del último elemento agregado. Cada que un elemento se agrega a la pila incrementamos en 1 dicho índice y agregamos el nuevo elemento al arreglo en la posición que la cima indica.

En el caso de un pop realizamos lo opuesto: regresamos el elemento que se encuentra en la posición de la cima y decrementamos su valor en 1.

Se podría también eliminar (volver nulo) el valor en dicha posición antes de regresarlo, pero de todas formas estaremos ocupando la misma cantidad de memoria por lo que la supuesta optimización no tendría sentido.

> A = Stack(5) # Nueva pila de 5 elementos



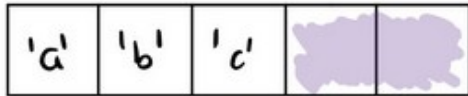
↑ cima = -1

> A.push('a')



↑ cima = 0

> A.push('c')



↑ cima = 2

> A.pop() # Regresa 'b'



↑ cima = 0

> A.push('b')



↑ cima = 1

> A.pop() # Regresa 'c'



↑ cima = 1

> A.push('z')



↑ cima = 2

Bibliografías

https://es.wikipedia.org/wiki/Estructura_de_datos

<https://www.tutorialesprogramacionya.com/cmasmasya/detalleconcepto.php?punto=32&codigo=163&inicio=30>

<https://tecnoinformatic.com/c-programacion/estructura-de-datos>

https://www.ecured.cu/Estructura_de_datos

<https://www.monografias.com/trabajos66/estructura-datos/estructura-datos.shtml>

<https://sites.google.com/site/soporteadistancia117360/home/diagrama-de-flujo>