**ARREGLOS (Arrays)**

Las operaciones que se pueden realizar con vectores durante el proceso de resolución de un problema son:

* Asignación
* Lectura / escritura
* Recorrido (acceso secuencial)
* Actualizar (añadir, borrar, insertar)
* Ordenación
* Búsqueda

**MATRICES BIDIMENSIONALES**

Los vectores examinados hasta ahora se denominan arrays unidimensionales y en ellos cada elemento se define por un índice.

Estos vectores son elementos de datos escritos por una secuencia. Sin embargo, existen grupos de datos que son representados mejor en forma de tabla o matriz con dos o más índices.

Un ejemplo de ellos es: tablas de distancias kilométricas entre ciudades, etc.

Se pueden definir tablas o matrices como arrays multidimensionales, cuyos elementos se pueden referenciar por dos, tres o más índices.

Los arrays no unidimensionales los dividiremos en dos grandes grupos:

* Arrays bidimensionales (2 dimensiones)
* Arrays multidimensionales (3 o más dimensiones)

El array bidimensional se puede considerar como un vector de vectores, es, por consiguiente, un conjunto de elementos, todos del mismo tipo, en el cual el orden de los componentes es significativo y en el que se necesita especificar dos índices para poder identificar cada elemento array.

Un índice no es suficiente para especificar un elemento de un array bidimensional.

Las operaciones que se pueden realizar con vectores durante el proceso de resolución de un problema son:

* Asignación
* Lectura / escritura
* Recorrido (acceso)
* Actualizar (añadir, borrar, insertar)

**LISTAS LIGADAS O ENLAZADAS (Linked lists)**

Una **lista** es aquella estructura que representa un numero contable de valores ordenados donde un mismo valor puede repetirse y considerarse un valor distinto a otro ya existente. Las listas son consideradas secuencias de valores.

Se pueden usar punteros (apuntadores) para implementar una lista enlazada. Los inconvenientes de las listas contiguas se eliminan con las listas enlazadas. Se pueden almacenar los elementos de una lista lineal en posiciones de memoria que no sean contiguas o adyacentes.

Un **puntero (apuntador)** es una variable cuyo valor es la dirección o posición de otra variable.

Una **lista enlazada, ligada o encadenada** es un conjunto de elementos en los que cada elemento contiene la posición o dirección del siguiente elemento de la lista.

Cada elemento de la lista enlazada debe tener al menos dos campos: un campo que tiene el valor del elemento y un campo (enlace, link) que contiene la posición del siguiente elemento, es decir, su conexión, enlace o encadenamiento.

Cada nodo contiene al menos un campo “pointer” apuntando a otro nodo.

Es necesario tener un pointer que apunte a la cabeza de la estructura para así saber por dónde comenzar.

La implementación de una lista enlazada depende del lenguaje. C, C++… utilizan simplemente como enlace una variable puntero (apuntador). Java no dispone de punteros, por consiguiente, resuelve el problema de forma diferente y almacena en el enlace la referencia al siguiente objeto nodo.

Las operaciones que normalmente se ejecutan con listas incluyen:

* Recuperar información de un nodo especifico (acceso a un elemento).
* Encontrar el nodo que contiene una información específica (localizar la posición de un elemento dado).
* Insertar un nuevo nodo en un lugar específico de la lista.
* Insertar un nuevo nodo con relación a una información particular.
* Borrar (eliminar) un nodo existente que contiene información específica.

Las estructuras de tipo contenedor se caracterizan principalmente por la forma particular de recuperación ordenada de los datos que soportan, y en los dos tipos principales de contenedores el orden de recuperación depende del orden de inserción.

Para este caso tenemos dos tipos principales de contenedores:

* Pilas (stack)
* Colas (queue)

**PILAS (stack)**

Una pila (stack) es un tipo especial de lista lineal en la que la inserción y borrado de nuevos elementos se realiza solo por un extremo que se denomina cima o tope (top).

La pila es una estructura con numerosas analogías en la vida real: una pila de platos, una pila de monedas, etc…

Dado que las operaciones de insertar y eliminar se realizan por un solo extremo (el superior), los elementos solo pueden eliminarse en orden inverso al que se insertan en la pila.

Las operaciones mas asociadas a las pilas son las siguientes:

* Push: insertar, poner o apilar un nuevo elemento a la pila de datos.
* Pop: sirve para sacar, quitar o des apilar un elemento de nuestra pila de datos.

Las pilas deben trabajarse con apuntadores y memoria dinámica. Aunque también se pueden declarar de forma estática.

**COLAS (queue)**

Las colas son otro tipo de estructura lineal de datos similar a las pilas, diferenciándose de ellas en el método de insertar y eliminar elementos.

Una cola es una estructura lineal de datos.

En las colas el elemento que entró primero sale también primero; por ello se le conoce como las listas FIFO (first-in, first-out), “primero en entrar, primero en salir”.

La diferencia con las pilas reside en el modo de entrada/salida de datos; en las colas las inserciones se realizan al final de las listas, no al principio.

Las colas se usan para almacenar datos que necesitan ser procesados según el orden de llegada.

Para representar una cola **Qu,** se puede definir mediante un vector, pero también se puede definir mediante una lista usando apuntadores.

Aunque las colas se pueden simular mediante un array y dos variables numéricas (frente y final), deberemos, si el lenguaje lo permite implementarlas mediante punteros.

En una cola las eliminaciones se realizarán por el extremo denominado frente y las inserciones por el final.

Algunas de las operaciones que se pueden realizar con las colas son las siguientes:

* Acceder al primer elemento de la cola
* Añadir un elemento al final de la cola
* Eliminar el primer elemento de la cola
* Vaciar toda la cola
* Verificar el estado de la cola: vacía o llena.

**ARBOLES**

El árbol es una estructura de datos fundamental en informática, muy utilizada en todos sus campos, porque se adapta a la representación natural de informaciones homogéneas organizadas y de una gran comodidad y rapidez de manipulación.

Esta estructura se encuentra en todos los campos de la informática, desde la pura algorítmica (métodos de clasificación y búsqueda…) a la compilación (arboles sintácticos para representar expresiones o producciones posibles de un lenguaje) o incluso los dominios de la inteligencia artificial, (arboles de juegos, arboles de decisiones, de resolución, etc).

Las estructuras de tipo árbol se usan principalmente para representar datos con una relación jerárquica entre sus elementos.

* Todos los arboles se componen de nodos y son estructuras no lineales.

Existe un tipo de árbol denominado árbol binario que puede ser implementado fácilmente.

Un árbol binario es un conjunto finito de cero o más nodos, tales que:

* Existe un nodo denominado raíz del árbol
* Cada nodo puede tener 0,1 o 2 subárboles, conocidos como subárbol izquierdo y subárbol derecho.

Dos arboles binarios se dicen que son similares si tienen la misma estructura, y son equivalentes si son similares y contienen la misma información.

Un árbol binario esta equilibrado si las estructuras de los dos subárboles de cada nodo del árbol se diferencian en una unidad como máximo.

Las operaciones que se pueden realizar con los árboles binarios son:

* Insertar un elemento
* Actualizar un elemento
* Eliminar un elemento
* Buscar un elemento