

Ejemplos de tipos de datos abstractos

Arrays

Es un conjunto de datos del mismo tipo almacenados en la memoria del ordenador en posiciones adyacentes. Sus componentes individuales se llaman elementos y se distinguen entre ellos por el nombre del array seguido de uno o varios índices o subíndices. Estos elementos se puede procesar, bien individualmente, determinando su posición dentro del array, bien como array completo. El número de elementos del array se especifica cuando se crea este, en la fase declarativa del programa, definiendo el número de dimensiones o número de índices del mismo y los límites máximo y mínimo que cada uno de ellos puede tomar, que llamaremos rango.

Ejemplo de la creación de un arreglo en **Javascript**: `var cars = ["Saab", "Volvo", "BMW"];`

Listas

Una lista lineal es un conjunto de elementos de un tipo de dato que pueden variar en número y donde cada elemento tiene un único predecesor y un único sucesor o siguiente, excepto el primero y último de la lista. Esta es una definición muy general que incluye los ficheros y vectores. Los elementos de una lista lineal se almacenan normalmente contiguos – un elemento detrás de otro – en posiciones consecutivas de la memoria. Las sucesivas entradas en una guía o directorio telefónico, por ejemplo, están en líneas sucesivas, excepto en las partes superior e inferior de cada columna, Una lista lineal se almacena en la memoria principal de una computadora en posiciones sucesivas de memoria; cuando se almacenan en cinta magnética, los elementos sucesivos se presentan en sucesión en la cinta. Esta asignación de memoria se denomina almacenamiento secuencial. Posteriormente se verá que existe otro tipo de almacenamiento denominado encadenado o enlazado. Las líneas así definidas se denominan contiguas. Las operaciones que se pueden realizar con listas lineales contiguas son:

- Insertar, eliminar o localizar un elemento.
- Determinar el tamaño – número de elementos – de la lista.
- Recorrer la lista para localizar un determinado elemento.
- Clasificar los elementos de la lista en orden ascendente o descendente.

- Unir dos o mas listas en una sola.
- Dividir una lista en varias sublistas.
- Copiar la lista.
- Borrar la lista.

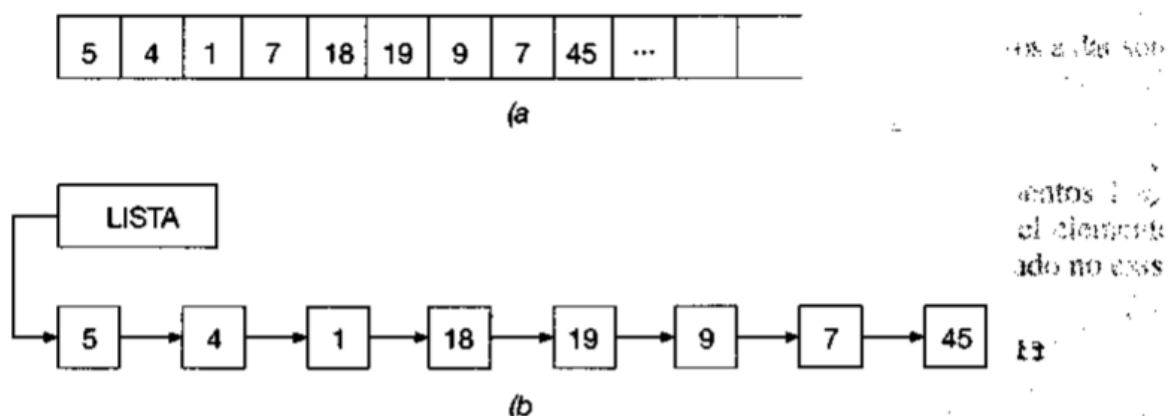
Una lista lineal contigua se almacena en la memoria de la computadora en posiciones sucesivas o adyacentes y se procesa como un array unidimensional. En este caso, el acceso a cualquier elemento de la lista y la adición de nuevos elementos es fácil; sin embargo, la inserción o borrado requiere un desplazamiento de lugar de los elementos que le siguen y, en consecuencia, el diseño de un algoritmo específico.

Ejemplo de la creación de una lista en **C#**: `List<string> lista = new List<string>();`

Listas enlazadas

Una lista enlazada o encadenada es un conjunto de elemento 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. Los elementos de una lista son enlazados por medio de los campos enlaces.

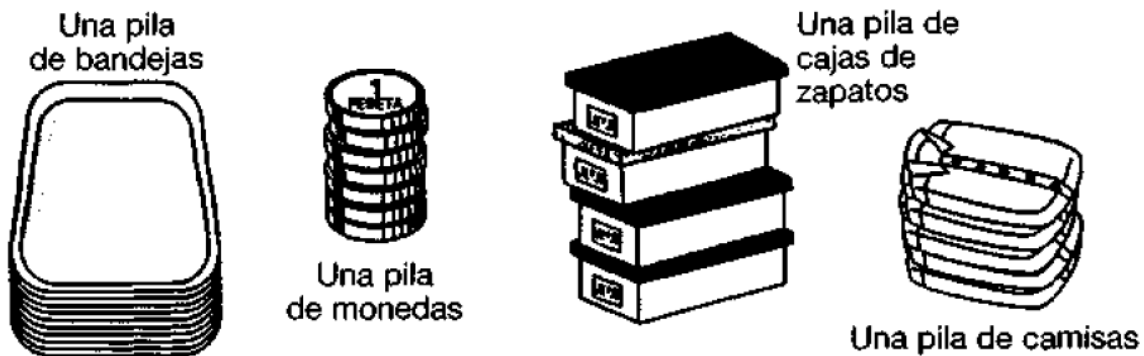
Ejemplo representación de una lista enlazada.



(a) array representado por una lista; (b) lista enlazada representada por una lista de enteros.

Pilas

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, una pila de cajas de zapatos, una pila de camisas, una pila de bandejas, 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. El último elemento que se pone en la pila es el primero que se puede sacar; por ello, a estas estructuras se les conoce por el nombre de LIFO (last – in, first –out, último en entrar, primero en salir). Las operaciones más usuales asociadas a las pilas son:

- “push” Meter, poner o apilar: operación de insertar un elemento en la pila.
- “pop” Sacar, quitar o desapilar: operación de eliminar un elemento de la pila.

Métodos para las pilas en **Java**.

empty(): Regresa valor booleano true si la pila se encuentra vacía.

push(item) : Introduce el objeto dentro de la pila.

pop(): Remueve el objeto del tope de la pila

peek(): Muestra el objeto del tope de la pila sin extraerlo.

search(item): Regresa un entero con la posición del objeto buscado o regresa un -1 si no se encuentra el objeto.

Ejemplo de pila en **Java**:

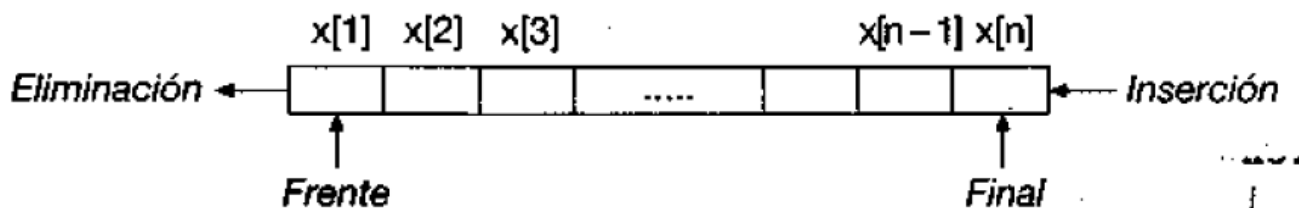
```
Stack pila =new Stack();  
pila.push("elemento 1");
```

Colas

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

Una cola (queue) es una estructura lineal de datos en la que las eliminaciones se realizan al principio de la lista, frente (front), y las inserciones se realizan en el otro extremo, final (rear). En las colas el elemento que entro el primero sale también el primero; por ello se conoce como listas FIFO (first- in, first – out, primero en entrar, primero en salir). Así, pues, la diferencia con las pilas reside en el modo de entrada/salida de datos; en las colas las inserciones se realizan al final de la lista, no al principio. Por ello las colas se usan para almacenar datos que necesitan ser procesados según el orden de llegada.

Ejemplo representación gráfica de colas.

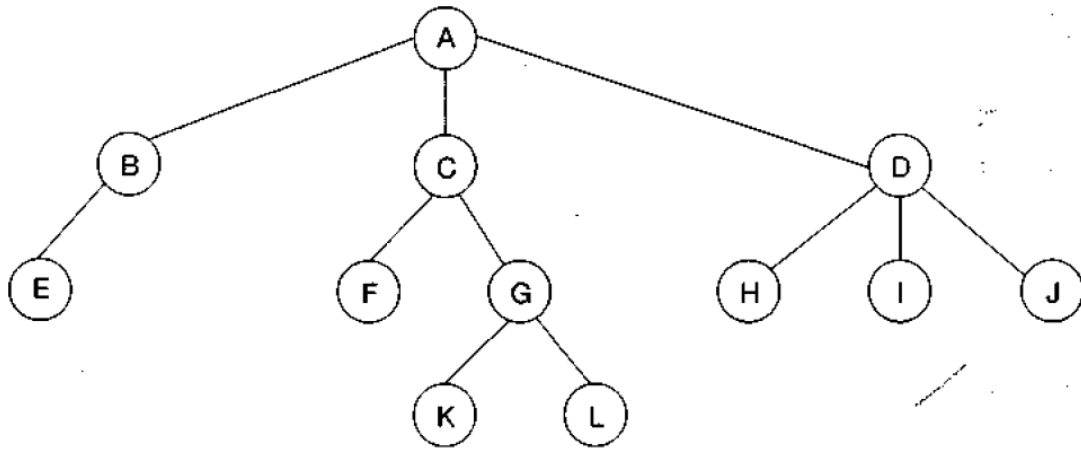


Arboles

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 dominios (campos) de la informática, desde la pura algorítmica (métodos de clasificación y búsqueda) a la compilación (arboles sintácticos para representar las expresiones o producciones posibles de un lenguajes) o incluso los dominios de la inteligencia artificial (arboles de juegos, arboles de decisiones, de resolución, etc.)

Las estructuras tipo árbol se usan principalmente para representar datos con una relación jerárquica entre sus elementos, como son árboles genealógicos, tablas, etc.

Ejemplo representación grafico de un árbol.



Bibliografía

Bottazzi, Cristian. (2017). Algoritmos y Estructuras de Datos. Argentina: UNL.

Luis Joyanes Aguilar. (2003). Fundamentos de programación – Algoritmos, Estructuras de datos y Objetos., de Mc Graw Hill