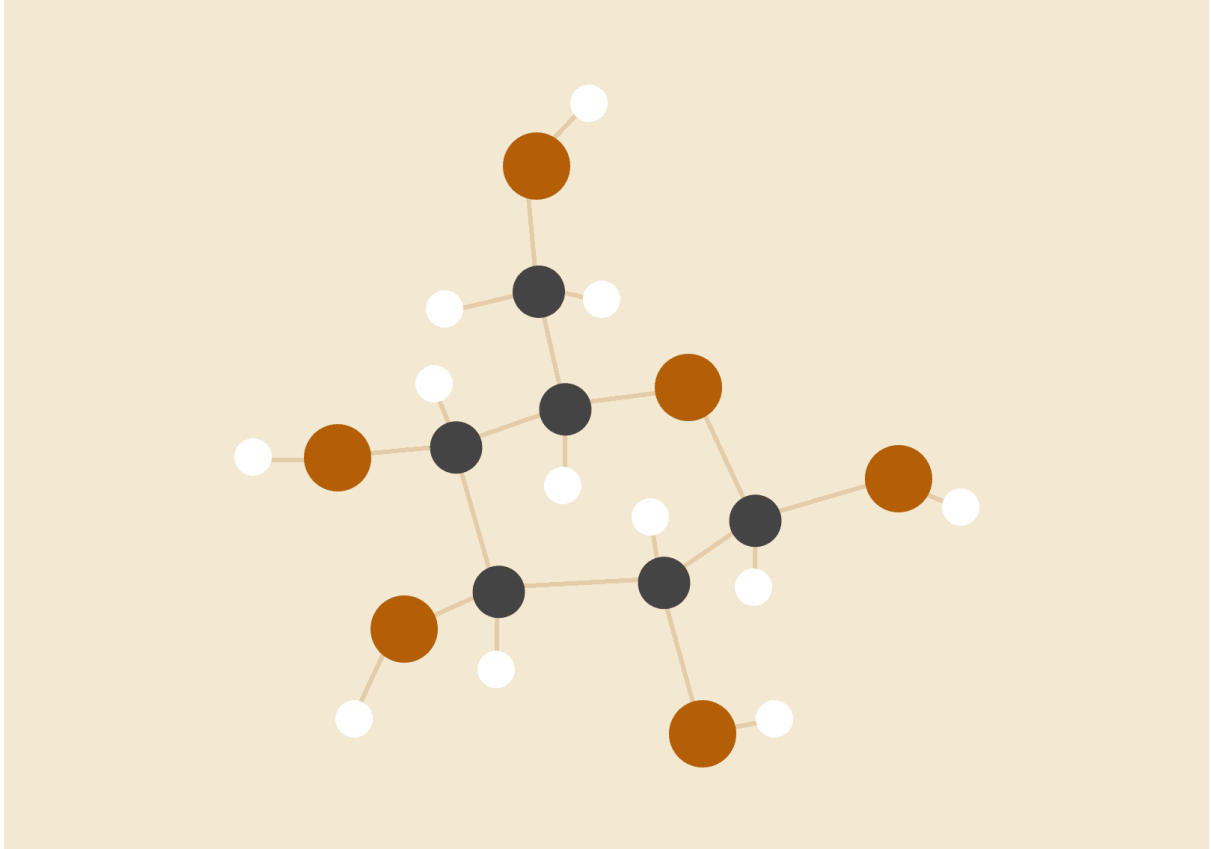


Algoritmos y Estructuras de Datos

Resumen de la cátedra impartida en la FCyT-UADER



GAREIS, Leonel

Rios Nuñez Marcos Elias

26 de Noviembre de 2019
Lic. en Sistemas de Información

Antes de comenzar, demos un breve repaso de algunas definiciones, tales como:

ALGORITMOS:

Es una secuencia de acciones o pasos que permite resolver un problema. Un mismo problema puede ser resuelto con distintos algoritmos.

ESTRUCTURAS DE DATOS:

Es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente. Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.

Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet. Por lo general, las estructuras de datos eficientes son clave para diseñar algoritmos eficientes.

Estas estructuras se dividen en:

Lineales:

Una estructura es lineal cuando un elemento solo puede tener un antecesor y un sucesor.

Dentro de esta categoría, podemos nombrar a:

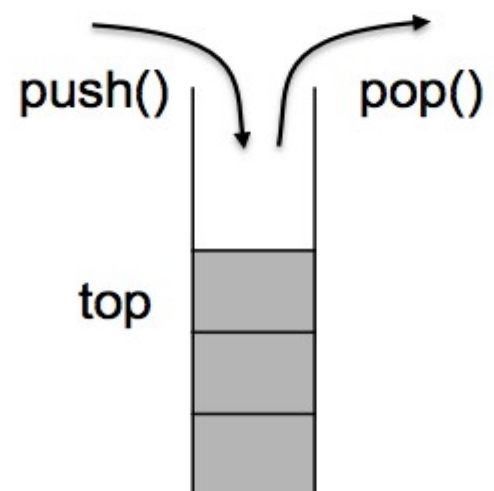
PILAS

Las pilas son otro tipo de estructura de datos lineales, las cuales presentan restricciones en cuanto a la posición en la cual pueden realizarse las inserciones y las extracciones de elementos.

Es un tipo especial de lista lineal en la que la inserción y borrado de elementos se realiza sólo por uno de los extremos. Como consecuencia, los elementos de una pila serán eliminados en orden inverso al que se insertaron.

Debido al orden en que se insertan y eliminan los elementos en una pila, también se le conoce como estructura de tipo LIFO (Last In, First Out), es decir: último en entrar, primero en salir.

En cuanto a su representación en memoria, se dice que las pilas no son estructuras de datos fundamentales, es decir, no están definidas como tales en los lenguajes de programación.



Las pilas pueden representarse mediante el uso de:

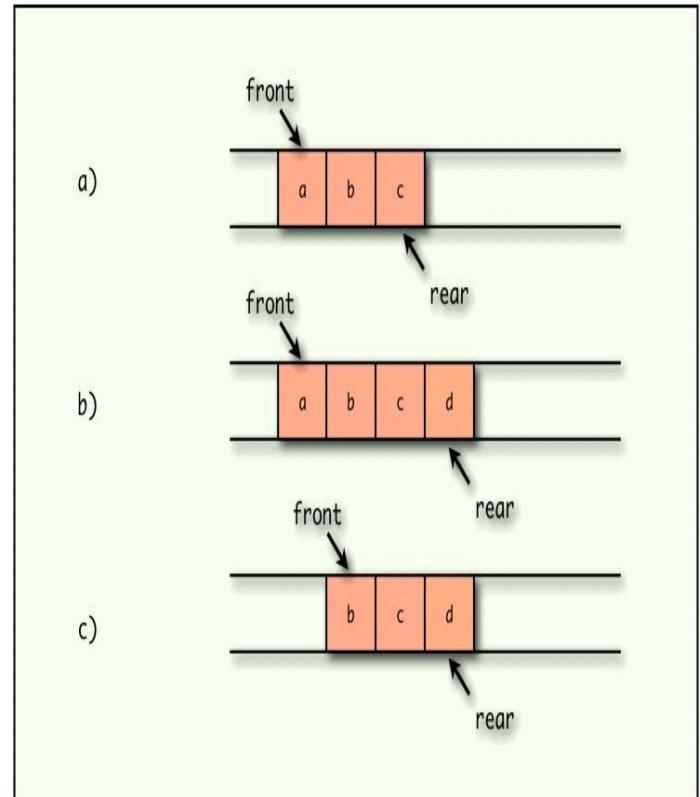
- Arreglos.
- Listas enlazadas.

COLAS

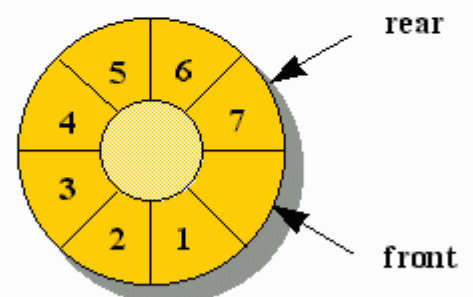
Es una estructura de datos en la cual la operación de adición se realiza por un extremo y la de eliminación por el otro.

Es una estructura de tipo **FIFO** (*First In, First Out*), es decir: **primero en entrar, primero en salir**.

Al igual que con las PILAS, las COLAS se pueden representar mediante Arreglos o Listas Simplemente Enlazadas.



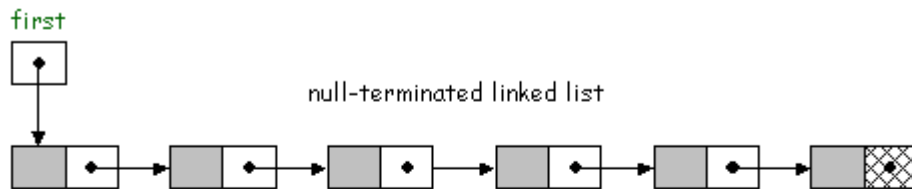
Para la representación con Arreglos, se utilizan las colas **Circulares**. Son útiles para controlar el tamaño de memoria (ya que lo define), una vez que llegó al valor debo borrar el primero para seguir agregando más. Mantengo hasta n elementos y el más viejo lo elimino para poder seguir agregando. Posee frente y fondo.



circular queue

Tanto las PILAS como las COLAS, son de acceso destructivo. Para poder ver el contenido de algún nodo, debo sacarlo de su estructura.

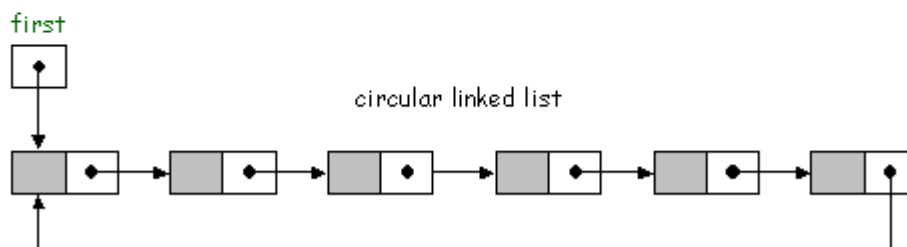
LISTAS



Este es un tipo de estructura **lineal** y **dinámica**: lineal porque a cada elemento le puede seguir solo otro elemento, dinámica porque su tamaño va cambiando en tiempo de ejecución.

Posee variaciones:

- **Lista Simplemente Enlazada**: los elementos de esta lista solo apuntan al siguiente
- **Lista Doblemente Enlazada**: los elementos de esta lista apuntan a su predecesor y su sucesor.
- **Lista Simplemente Enlazada Circular**: similar a la LSE con la diferencia que el último elemento apunta al primero.
- **Lista Doblemente Enlazada Circular**: similar a la LDE con la diferencia que el último elemento apunta al primero y viceversa.



Ejemplo de Lista Circular Simple

Una lista, puede ser una **MULTILISTA** (una lista de lista). Es decir una estructura compleja que utiliza 2 tipos de lista (Lista Principal y Lista Auxliar), para definir una estructura multinivel. Dicho de otra manera es una lista principal, simplemente enlazada, donde cada nodo de esta lista apunta además, a una lista auxiliar, también simplemente enlazada.

En las estructuras de datos **NO LINEALES**, cada elemento puede tener más de un antecesor y/o sucesor.

ÁRBOLES

Son Estructuras de Datos **no lineales y dinámicas**.

En general un Árbol se puede definir como una estructura jerárquica aplicada sobre una colección de elementos conocidos como nodos, donde uno de ellos es conocido como raíz. Estas ED son a su vez estructuras recursivas, ya que cada nodo interno forma un árbol, y un nodo de ese subárbol forma otro, y así sucesivamente.

Algunas definiciones que debemos manejar, son:

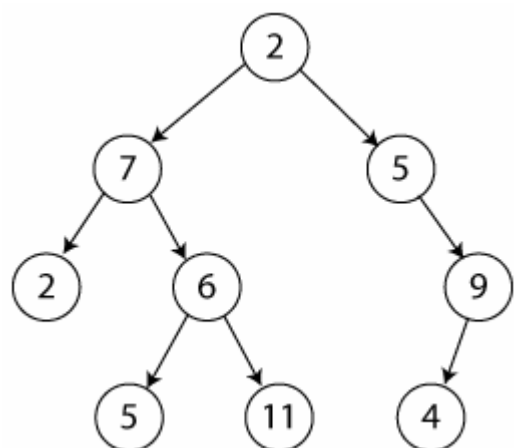
- **Longitud de camino:** cantidad de ramas desde la raíz hasta el nodo deseado
- **Altura de nodo:** cantidad de nodos desde la raíz hasta el nodo hoja más alejado
- **Profundidad/Nivel:** cantidad de nodos desde la raíz hasta el nodo
- **Nodos hermanos:** aquellos que poseen el mismo nodo predecesor (o padre)
- **Orden:** cantidad de sucesores que puede tener un mismo nodo
- **Grado de nodo:** cantidad de sucesores que posee.

Acá podemos destacar lo siguiente:

- + **Árboles Llenos:** todos los nodos hojas están al mismo nivel y cada nodo (excepto las hojas) posee la máxima cantidad de hijos.
- + **Árboles Completos:** todos los nodos (excepto las hojas) poseen la máxima cantidad de hijos
- + **Árboles Degenerados:** cada nodo (incluyendo la raíz) posee un único sucesor. Su figura se asemeja a la de una Lista SE

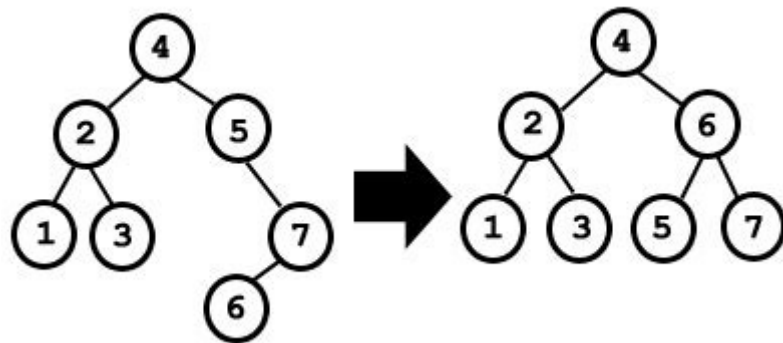
- **Binarios de Búsqueda:**

Son árboles de orden 2 donde, dado un nodo no hoja, su subárbol izquierdo es menor que el propio nodo, y el subárbol derecho es mayor



- **AVL:**

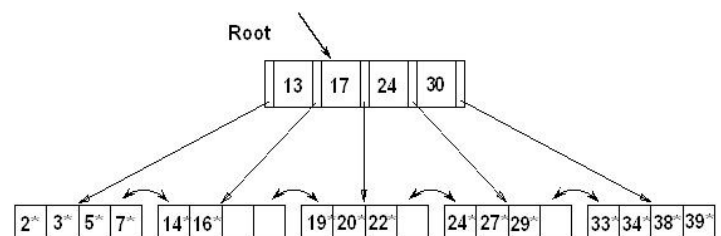
Estos árboles (al igual que los ABB) son de búsqueda, y poseen la particularidad de ser auto-balanceables. Los árboles AVL están siempre equilibrados, de tal modo que la altura del subárbol izquierdo no difiere en más de una unidad de la altura del subárbol derecho. Para ello utiliza las “rotaciones AVL” y un Factor de Equilibrio (que es un elemento adicional que se usa en cada nodo para determinar si se mantiene o no la condición de AVL. Posee tres estados: balanceado, desbalanceado con carga a la derecha/izquierda)



Rotación Simple: el exceso en profundidad se da en un sentido, el crecimiento es a izquierda o a derecha. La raíz del sub-árbol donde se produjo el desbalance sube un nivel, y a partir de aquí el árbol se reestructura como un árbol binario de búsqueda.

Rotación doble: se da para el caso en el que el desbalance no se produce en un sentido único. La raíz del sub-árbol en el que se produjo el exceso sube dos niveles, se efectúan dos rotaciones.

- **B:** es una estructura multimodal de orden “m” que cumple con las siguientes condiciones
 - + Todas las hojas están al mismo nivel
 - + Todos los nodos internos tienen a lo sumo M hijos y como mínimo m/2 hijos.
 - + El número de claves en cada nodo interno es m-1



- **La búsqueda** es similar a la de los árboles binarios. Se empieza en la raíz, y se recorre el árbol hacia abajo, escogiendo el sub-nodo de acuerdo a la posición relativa del valor buscado respecto a los valores de cada nodo. Típicamente se utiliza la búsqueda binaria para determinar esta posición relativa.

- **Las inserciones** se hacen en los nodos hoja.
Realizando una búsqueda en el árbol, se halla el nodo hoja en el cual debería ubicarse el nuevo elemento.
Si el nodo hoja tiene menos elementos que el máximo número de elementos permitidos ($n-1$), entonces hay lugar para uno más. Inserte el nuevo elemento en el nodo, respetando el orden de los elementos.
De otra forma, el nodo debe ser dividido en dos nodos.

La división se realiza de la siguiente manera:

1. Se escoge el valor medio entre los elementos del nodo y el nuevo elemento.
2. Los valores menores que el valor medio se colocan en el nuevo nodo izquierdo, y los valores mayores que el valor medio se colocan en el nuevo nodo derecho; el valor medio actúa como valor separador.
3. El valor separador se debe colocar en el nodo padre, lo que puede provocar que el padre sea dividido en dos, y así sucesivamente.

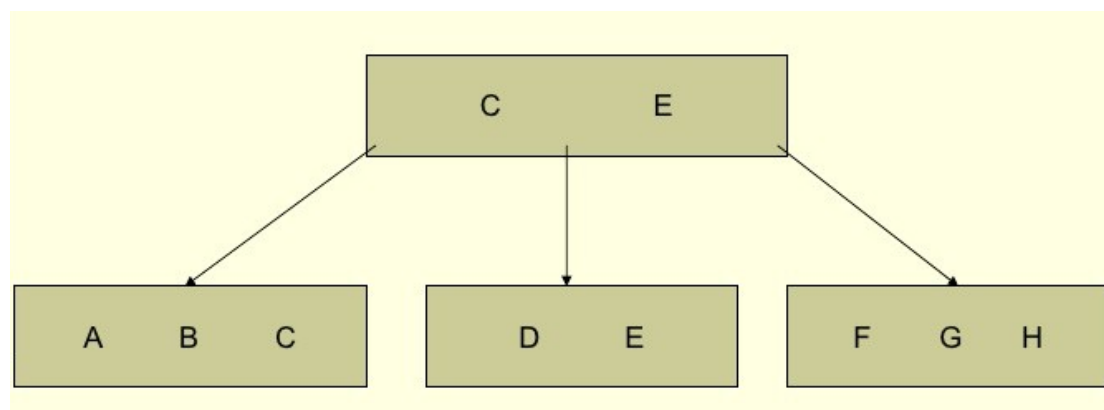
Eliminación

Es una operación más complicada, consiste en eliminar o quitar la clave sin violar los principios del árbol B.

Si la clave a borrar se encuentra en una hoja simplemente se suprime, si no debe bajarse la clave adyacente de la página antecedente y sustituir esta clave por la que se encuentra más a la derecha del sub-árbol izquierdo o por la que se encuentra más a la izquierda del sub-árbol derecho.

- **B+**: similar al B, con la diferencia de que cuando se divide un nodo hoja se deja una copia del padre en el nodo izquierdo.

Además cada nodo hoja apunta a su hermano o primo.



Principales características de los árboles B+ de orden m son:

- La raíz almacena como mínimo un dato y como máximo $m-1$ datos.
- La página raíz tiene como mínimo dos descendientes.
- Las páginas intermedias tienen como mínimo $(m-1)/2$ (Parte entera) datos.
- Las páginas intermedias tienen como máximo $m-1$ datos.
- Todas las páginas hojas tienen la misma altura
- La información se encuentra ordenada.
- Toda la información se encuentra almacenada en las páginas hoja, por lo que en las páginas internas se puede duplicar la claves.

Inserción en un árbol B+:

es similar a la del árbol B se diferencia en el momento que una página deja de cumplir la condición del número de datos almacenados. Para realizar se debe subir una copia de la clave mediana de los datos del nodo a la página padre, solo se duplica la información cuando la clave que sube es de una página hoja.

Los pasos a seguir para una inserción son los siguientes:

1. Se ubica en la página raíz.
2. Se evalúa si es una página hoja
 - 2.1. Si la respuesta es afirmativa, se evalúa si no sobrepasa los límites de datos.
 - 2.1.1. Si la respuesta es afirmativa, entonces se procede a insertar el nuevo valor en lugar del correspondiente.
 - 2.1.2. Si la respuesta es negativa, se divide la página en dos, se sube una copia de la mediana a la página padre, si la página padre se encuentra llena se debe de partir igual y así el mismo proceso hasta donde sea necesario, si este proceso llega hasta la raíz la altura del árbol aumenta en uno.
 - 2.2. Si no es hoja, se compara el elemento a insertar con cada uno de los valores almacenados para encontrar la página descendiente donde proseguir la búsqueda. Se regresa al paso 1.

Eliminación:

La operación de eliminación en árboles-B+ es más simple que en árboles-B. Esto ocurre porque las claves a eliminar siempre se encuentran en las páginas hojas. En general deben distinguirse los siguientes casos:

1. Si al eliminar una clave, la cantidad de llaves queda mayor o igual que $\lceil m/2 \rceil$ entonces termina la operación. Las claves de los nodos raíz o internos no se modifican por más que sean una copia de la clave eliminada en las hojas.
2. Si al eliminar una clave, la cantidad de claves queda menor que $\lceil m/2 \rceil$ entonces debe realizarse una redistribución de claves, tanto en el índice como en las páginas hojas

Barridos: Es un algoritmo que devuelve la secuencia de todos los nodos contenidos en un árbol. Todo árbol

- considerado como estructura es accedido por la raíz.
- Hay cuatro criterios distintos para el barrido de árboles, tres de los cuales son aplicables sólo en árboles binarios, el cuarto barrido (por niveles) es aplicable a cualquier árbol: preorden, postorden, simétrico y por niveles.

Los recorridos de un árbol en los órdenes previo y posterior permiten determinar los antecesores de un nodo.

○

- + **PREORDEN:** primero se procesa la raíz, seguido del subárbol izquierdo y por último el subárbol derecho (R,I,D)
- + **INORDEN:** primero se procesa el subárbol izquierdo, seguido de la raíz y por último el sub-árbol derecho (I,R,D)
- + **POSTORDEN:** primero se procesa el subárbol izquierdo, seguido del subárbol derecho y por último la raíz (I,D,R)
- + **Recorrido por niveles:** se muestra la raíz, y se sigue hacia abajo de

izquierda a derecha

Arboles N-arios:

Es una estructura de datos donde cada nodo posee un numero indeterminado de hijos. es una estructura recursiva, y corresponde a la generalización de un árbol binario de cuyos nodos pueden desprenderse múltiples arboles binarios. Las reglas que aplican a los arboles binarios se pueden transpolar a los arboles n-arios así como los consejos base.

Un árbol n-ario puede tomarse como un árbol de n elementos asociados a cada uno de sus componentes. se pueden encontrar 3 tipos de recorridos para este tipo de arbol:

- INORDEN
- PREORDEN
- POSTORDEN

Posee los mismos conceptos que un árbol binario:

1. Nodo
2. Raíz
3. Hoja
4. Camino
5. Rama
6. Altura
7. Peso

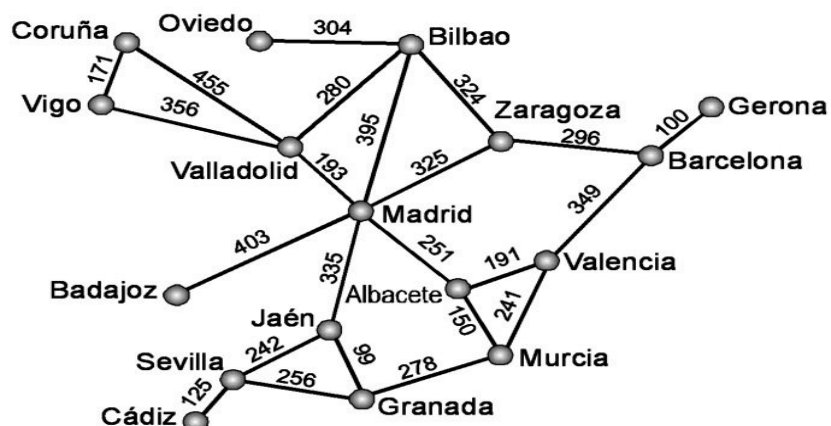
Las 3 operaciones que se pueden hacer a partir de estos arboles son:

- Adición
- Sustracción
- Búsqueda

GRAFOS

Es una ED que está formada por un **conjunto de nodos** (denominado P, y simboliza los objetos) y un **conjunto de arcos** (denominado R, simboliza las relaciones entre los objetos).

Un grafo está definido de la siguiente manera: $G=(P,R)$



$P = \{x/x \text{ es un nodo del modelo}\}$; ej: $P = \{\text{elem1, elem2, ..., elem n}\}$

$R = \{(x,y)/(x,y) \in P \wedge \text{"x está relacionado con y"}\}$ ej: $R = \{(x1,y1), (x2,y2), ..., (xn,yn)\}$

Funciones de asignación: Características o propiedades de los nodos o de las relaciones.

Función de asignación a nodo: Caracterizan al conjunto de nodos

Función de asignación a arco: Caracterizan a las relaciones.

Sabiendo esto, podemos definir lo siguiente:

- **Subgrafo:** es un grafo incluido en otro.
- **PASO:** $\rho(x,z)$ es la secuencia de $\langle y_0, y_1, \dots, y_n \rangle$, con $n \geq 0$ que cumple con las siguientes condiciones:
 - $x = y_0 \quad z = y_n$ (todo paso tiene origen y final)
 - $y_{i-1} \neq y_i$ (no hay pasos consecutivos iguales)
 - $(y_{i-1}, y_i) \in R$ con $1 \leq i \leq n$
- **LONGITUD DE PASO:** $|\rho(x,z)|$ = cantidad de arcos entre x y z
- **CAMINO:** $C(x,z)$ es la secuencia $\langle y_0, y_1, \dots, y_n \rangle$, con $n \geq 0$ que cumple las siguientes condiciones:
 - $x = y_0 \quad z = y_n$
 - $y_{i-1} \neq y_i$
 - $(y_{i-1}, y_i) \wedge (y_i, y_{i+1}) \in R$ con $1 \leq i \leq n$
- **LONGITUD DE CAMINO:** $|C(x,z)|$ = cantidad de arcos en ese camino
- **CICLO:** $|\rho(x,x)| \geq 2$ (longitud de paso entre x y x)
- **CIRCUITO:** $|C(x,x)| \geq 2$ (cantidad de nodos entre x y x)
- **LOOP:** $|\rho(x,x)| = 0$
- **CONJUNTO IZQUIERDO (LEFT):** es el conjunto de nodos desde los cuales se puede llegar al nodo considerado con longitud de paso igual a 1 (es decir, directamente)

$$L(x) = \{y/y \in P \text{ y } (y,x) \in R\}$$

- **CONJUNTO DERECHO (RIGHT):** es el conjunto de nodos a los cuales se puede llegar desde el nodo considerado, con longitud de paso igual a 1.

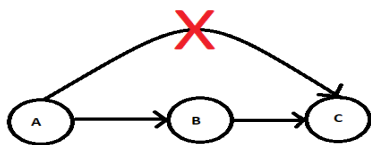
$$R(x) = \{y/y \in P \text{ y } (x,y) \in R\}$$

- **GRADO DE ENTRADA** = $|L(x)|$
- **GRADO DE SALIDA** = $|R(x)|$
- **GRADO DE ENTRADA DE UN GRAFO:** grado del nodo que posee el mayor grado de entrada del grafo
- **GRADO DE SALIDA DE UN GRAFO:** grado del nodo que posee el mayor grado de salida del grafo
- **CONJUNTO IDEAL IZQUIERDO:** es el conjunto de nodos desde los cuales se puede llegar al nodo considerado, directa o indirectamente.
- **CONJUNTO IDEAL DERECHO:** es el conjunto de nodos a los cuales puedo llegar

desde el nodo considerado, directa o indirectamente.

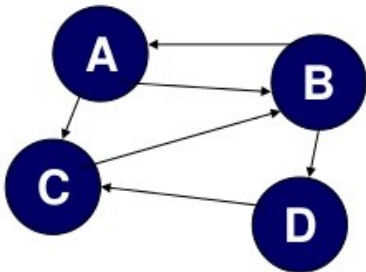
- **CONJUNTO MINIMAL:** conjunto de puntos a los cuales no llega ningún arco. El grado de entrada es 0
- **CONJUNTO MAXIMAL:** conjunto de puntos desde los cuales no sale ningún arco. El grado de salida es 0
- **MÍNIMO:** único nodo en el grafo al cual no le llegan arcos. Se da cuando el conjunto minimal está formado por un único nodo.
- **MÁXIMO:** único nodo en el grafo desde el que no salen arcos. Se da cuando el conjunto maximal está formado por un único nodo.
- **Clausura Transitiva:** es un grafo en el que los nodos son los mismos pero la relación resulta de generar todos los arcos posibles con respecto a la relación original.

Tipos de Grafos



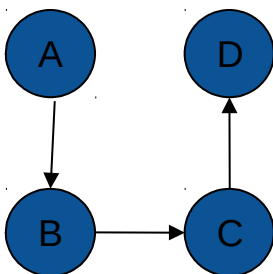
Grafo Básico: se dice que un grafo es simple si:

- Está libre de loops
- $\forall x, y \in P, \text{ si } \exists |r(x, y)| \geq 2 \rightarrow (x, y) \notin R$



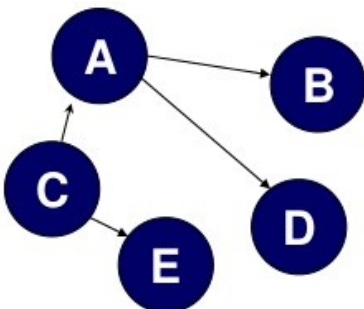
Grafo Cíclico:

- Contiene al menos 1 ciclo
- $\forall x \in P, \exists |r(x, x)| \geq 2$



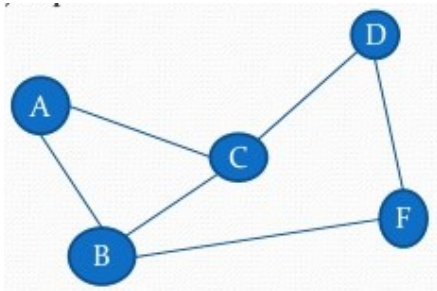
Grafo Lineal

- $\exists x \in P / L(x) = 0$ (hay al menos un nodo al que no le llegan arcos)
- $\forall y \in P, |L(y)| \leq 1, |R(y)| \leq 1$ (grado de entrada y salida ≤ 1)
- G debe ser conexo



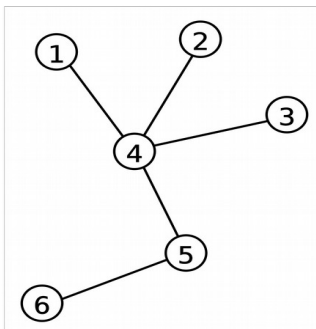
Grafo Acíclico

- No posee ciclos
- $\forall x \in P, \nexists |r(x, x)| \geq 2$



Grafo Conexo

- No tiene puntos aislados
- Existe un camino entre dos nodos cualesquiera
- **Todos sus pares de nodos están conectados**



Grafo Finito

Un grafo $G = (P, R)$ se dice finito si P es un conjunto finito. cantidad máxima de flechas que puede haber en un grafo G finito $= n^2 - n$
 Nota: n es la cantidad de nodos o puntos del modelo.

Grafos Iguales:

Dos grafos son iguales si y sólo si uno es copia del otro.

Grafos Isomorfos

Sea $G_1=(P_1,R_1)$ y $G_2=(P_2,R_2)$, se dice que son isomorfos si:

- $\exists \varphi: P_1 \rightarrow P_2 / \forall x, y \in P_1, (x, y) \in R_1 \Leftrightarrow (\varphi(x), \varphi(y)) \in R_2 \wedge \varphi(x), \varphi(y) \in P_2$

grafos que tienen la misma estructura pero sus nodos (componentes) son distintos.

Grafo G	Grafo H	Un isomorfismo entre G y H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

Hipergrafo

Es un grafo que contiene un conjunto de puntos y más de una relación definida en dicho conjunto

Esquema resultante: $H=(P, R, R', \dots, R(n), f_1, f_2, \dots, f_k, g_1, g_2, \dots, g_m, h_1, h_2, \dots, h_p, \dots)$

Un Árbol es un grafo que:

1. es conexo y conectado.
2. $|P|=|R| + 1$.
3. cualquier arco adicional crea un Circuito.
4. Libre de loops.
5. El camino entre un nodo y otro es único.

Grafos Irrestringidos

Son grafos que definen su estructura a partir del modelo de la realidad que se desea implementar. Esto hace que no se conozca a priori la cantidad de nodos ni la cantidad de arcos necesarios, ni las relaciones, etc.

Los grafos pueden ser representados mediante un array bidimensional (conocido como **matriz de adyacencia**), o bien una representación dinámica, mediante una estructura multi enlazada (conocida como **lista de adyacencia**) y **Lista de Adyacencia Completa**.

La elección depende del tipo de grafo y de las operaciones que se vayan a realizar sobre los nodos y los arcos.

- **Matriz de Adyacencia:** el grafo se representa mediante una matriz $A_{N \times N}$ (siendo N el número de nodos), tal que todo elemento a_{ij} puede tomar los valores:
 - 0 si no hay arco (V_i, V_j)
 - 1 si hay un arco (V_i, V_j)

Estableciendo las relaciones se obtiene el número de arcos.

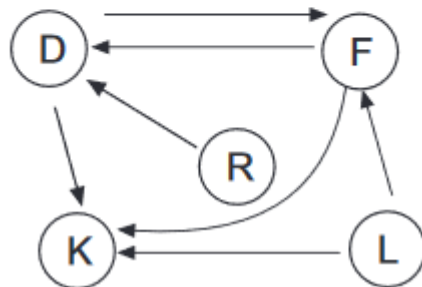
Siempre se lee de Columna a Fila.

Representa un listado de nodos. Indica mediante alguna condición de la matriz, cuáles son los nodos adyacentes.

Características:

- *N debe conocerse de antemano, es decir se usan Grafos poco volátiles.*
- *Permite ejecutar consultas rápidas.*
- *Estas matrices no se almacenan, se alojan en memoria.*
- *Es rígida tengo que saber con anticipación la cantidad de datos, de lo contrario podría definir una mega matriz, solo para tener o generar muchos espacios subocupados.*
- *Tiene mucha dispersión de datos*
- *ocupa muchísimos espacios para guardar alguna información.*
- *Cuando la cantidad de arcos es significativamente menor que la cantidad de nodos, resulta contraproducente Ej: mil nodos, 5 arcos. Es un gran trabajo para obtener pocos resultados.*

$$A = \begin{vmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{vmatrix}$$

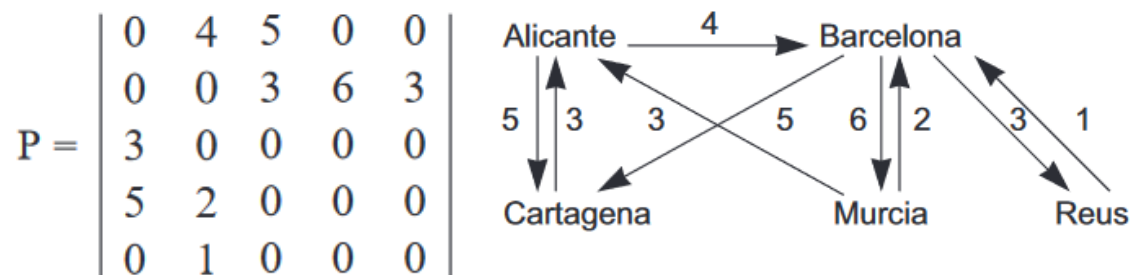


Donde

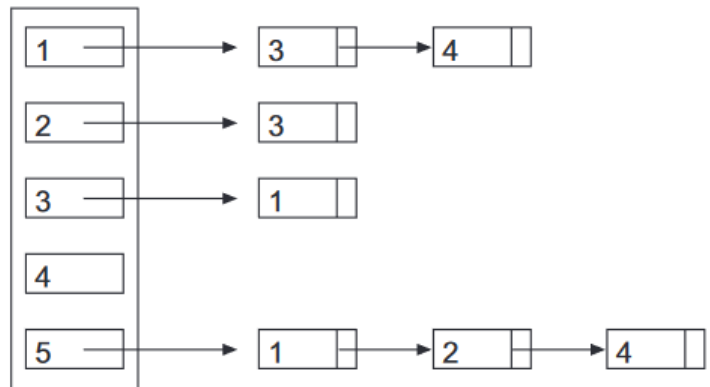
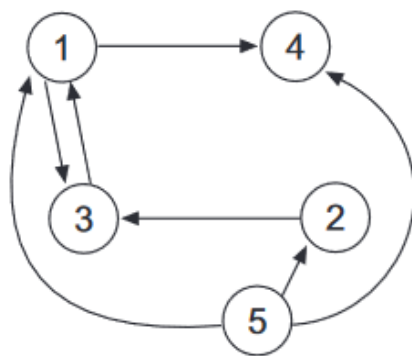
$P = \{D, F, K, L, R\}$

$R = \{(D,F);(D,K);(F,D);(F,K);(L,F);(L,K);(R,D)\}$

En el caso de que se trate de arcos con un valor se hace uso de una matriz valorada



- **Lista de Adyacencia:** es una estructura multi enlazada formada por una “tabla directorio” donde cada nodo representa un elemento (vértice) del grafo, del que emerge una lista enlazada con todos sus vértices adyacentes. las listas de adyacentes no son ordenadas.



Preguntas Frecuentes

¿Una lista es un Grafo? - si, se considera que la lista es un Grafo con restricción.

¿Un grafo puede ser una lista? - No

¿Un árbol es un grafo? - si, es un grafo con restricciones definidas

¿Un grafo es un árbol? - si

Se puede interpretar que un árbol B+ también es un grafo. El grafo no tiene restricciones y todo lo que hemos estado viendo han sido grafos con restricciones.

¿Una multilista es un Hipergrafo?

Considerando la definición de Hipergrafo, la cual dice que para un mismo conjunto de puntos pueden existir varias relaciones y a su vez funciones de asignación a nodos y funciones de asignación a arcos, y la definición de multilista, que nos dice que es una lista que contiene varias listas en su interior; podemos decir que una multilista es un Hipergrafo.

¿Las listas doblemente enlazadas son multilista?

No, ya que las listas doblemente enlazadas contienen dos campos de enlace y se puede avanzar y retroceder

en la lista, precisamente porque contiene puntero al antecesor y al sucesor, cosa que no ocurre con las multilista, que contienen listas como información y no se puede retroceder sobre la lista.

¿Una lista doblemente enlazada es una estructura lineal?

Una lista doblemente enlazada no es una estructura lineal, ya que sus elementos tienen más de un antecesor y más de un sucesor.