UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES
ESTRUCTURA DE ARCHIVOS
SECCIÓN D01

TAREA 03: ÁRBOLES

MIGUEL ÁNGEL GUERRERO SEGURA RAMÍREZ
PROFESOR

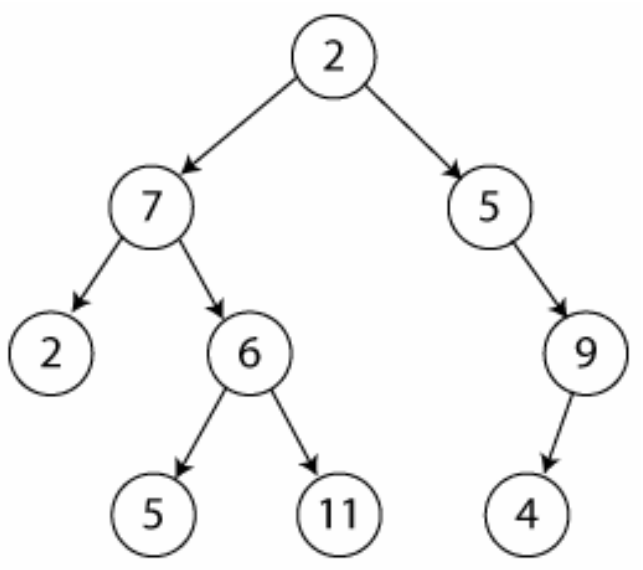
PARTIDA RAMÍREZ EDSON CARLOS 209539034
ALUMNO 4º INGENIERÍA EN COMPUTACIÓN

GUADALAJARA, JALISCO
ABRIL DE 2014

ÁRBOLES

DEFINICIÓN

Un árbol es una estructura de datos ampliamente usada que imita la forma de un árbol (un conjunto de nodos conectados). Un ***nodo*** es la unidad sobre la que se construye el árbol y puede tener cero o más ***nodos hijos*** conectados a él. Se dice que un nodo ***a*** es ***padre*** de un nodo ***b*** si existe un enlace desde a hasta b (en ese caso, también decimos que ***b*** es ***hijo*** de ***a***). Sólo puede haber un único nodo sin padres, que llamaremos ***raíz***. Un nodo que no tiene hijos se conoce como ***hoja***. Los demás nodos (tienen padre y uno o varios hijos) se les conoce como ***rama***.



El enlace entre nodos padres y nodos hijos suele estar dado por punteros (apuntadores de dirección de memoria), por ello otra característica que normalmente tendrán los árboles es que todos los nodos contienen el mismo número de punteros, es decir, se usará la misma estructura para todos los nodos del árbol. Esto hace que la estructura sea más sencilla, y por lo tanto también los programas para trabajar con ellos.

Tampoco es necesario que todos los nodos hijos de un nodo concreto existan. Es decir, que pueden usarse todos, algunos o ninguno de los punteros de cada nodo. Un árbol en el que en cada nodo o bien todos o ninguno de los hijos existe, se llama árbol completo.

Otros conceptos que deben conocerse respecto a la estructura de un árbol son los siguientes:

- **Orden:** Es el número potencial de hijos que puede tener cada elemento de árbol. De este modo, diremos que un árbol en el que cada nodo puede apuntar a otros dos es de orden dos, si puede apuntar a tres será de orden tres, etc. El árbol de ejemplo es de Orden 2.
- **Grado:** Es el número de hijos que tiene el elemento con más hijos dentro del árbol. En el árbol del ejemplo, el grado es dos, ya que tanto 7 como 6 y 2 tienen dos hijos, y no existen elementos con más de dos hijos.

- **Nivel:** Se define para cada elemento del árbol como la distancia a la raíz, medida en nodos. El nivel de la raíz es cero y el de sus hijos uno. Así sucesivamente. En el ejemplo, el nodo 7 tiene nivel 1, el nodo 6 tiene nivel 2, y el nodo 11, nivel 3.
- **Altura:** La altura de un árbol se define como el nivel del nodo de mayor nivel. Como cada nodo de un árbol puede considerarse a su vez como la raíz de un sub-árbol, también podemos hablar de altura de ramas. El árbol del ejemplo tiene altura 3, la rama 7 tiene altura 2, la rama 6 tiene altura 1, la 5 cero, etc.

OPERACIONES BÁSICAS SOBRE ÁRBOLES

Debido a que existen muchos tipos de árboles, las operaciones que se pueden efectuar sobre éstos pueden variar significantemente entre cada tipo; sin embargo, tratando de hacer un enfoque generalizado sobre los árboles es posible encontrar ciertas operaciones en común:

- **AÑADIR O INSERTAR ELEMENTOS:** Consiste en agregar nodos al árbol que podrán ser nodos hijos, nodos padre, nodos ramas o nodos hojas según sea el caso.
- **BUSCAR O LOCALIZAR ELEMENTOS:** Una vez añadidos ciertos elementos en el árbol es necesario establecer un algoritmo que permita encontrar uno en específico.
- **BORRAR ELEMENTOS:** Así mismo es necesario poder eliminar elementos del árbol que ya no sean útiles, esta acción es un tanto especial pues requiere en muchas ocasiones la reestructuración de alguna parte del árbol o en el peor de los casos al árbol entero.
- **MOVERSE A TRAVÉS DEL ÁRBOL:** Moverse a través del árbol significará poder visualizar los elementos contenidos en este sin realizarles alguna modificación.
- **RECORRER EL ÁRBOL COMPLETO:** Esta es una variante de la operación anterior que involucrará a absolutamente todos los nodos del árbol desde la raíz hasta el nodo más profundo.

RECORRIDO DE UN ÁRBOL

Se pone especial énfasis en esta operación debido a que es una operación necesaria para otras operaciones como el eliminar un nodo, o buscar un nodo. Cabe destacar que debido a la naturaleza del árbol este puede ser recorrido de tres maneras diferentes: En PostOrden, InOrden y en PreOrden. Cualquiera de las tres permite recorrer el árbol completo de ser necesario. Lo que diferencia los distintos métodos de recorrer el árbol no es el sistema de hacerlo, sino el momento que elegimos para procesar el valor de cada nodo con relación a los recorridos de cada una de las ramas.

- **RECORRIDO PRE-ORDEN:** En este tipo de recorrido, el valor del nodo se procesa antes de recorrer las ramas. En el árbol de ejemplo si se decidiera mostrar el valor de los nodos según el recorrido en Preorden éstos se mostrarían de la siguiente manera: 2, 7, 2, 6, 5, 11, 5, 9, 4.

- **RECORRIDO IN-ORDEN:** En este tipo de recorrido, el valor del nodo se procesa después de recorrer la primera rama y antes de recorrer la última. Esto tiene más sentido en el caso de árboles binarios, y también cuando existen ORDEN-1 datos, en cuyo caso procesaremos cada dato entre el recorrido de cada dos ramas (este es el caso de los árboles-b). En el árbol de ejemplo si se decidiera mostrar el valor de los nodos según el recorrido en Inorden éstos se mostrarían de la siguiente manera: 2, 7, 5, 6, 11, 2, 5, 9, 4.
- **RECORRIDO POST-ORDEN:** En este tipo de recorrido, el valor del nodo se procesa después de recorrer todas las ramas. En el árbol de ejemplo si se decidiera mostrar el valor de los nodos según el recorrido en Postorden éstos se mostrarían de la siguiente manera: 2, 5, 11, 6, 7, 4, 9, 5, 2.

ELIMINACIÓN EN UN ÁRBOL

En un árbol hay dos posibles casos para la eliminación de un nodo: Cuando este sea Hoja, o bien cuando este sea Rama. Hay diferencias para cada caso, de esta forma en un árbol genérico, es decir de ningún tipo, se procede de la siguiente manera:

Cuando el nodo a eliminar es un nodo hoja hay que hacer lo siguiente:

1. Buscar el nodo padre del nodo que queremos eliminar.
2. Buscar el puntero del nodo padre que apunta al nodo que queremos borrar.
3. Liberar el nodo.
4. Redirigir el puntero del nodo padre hacia null para que deje de apuntar al nodo hoja eliminado.

Cuando el nodo a borrar no sea un nodo hoja, diremos que hacemos una "poda", y en ese caso eliminaremos el subárbol cuya raíz es el nodo a borrar. Se trata de un procedimiento recursivo, aplicamos el recorrido PostOrden, y el proceso será borrar el nodo.

El procedimiento es similar al de borrado de un nodo:

1. Buscar el nodo padre del nodo que queremos eliminar.
2. Buscar el puntero del nodo padre que apunta al nodo que queremos borrar.
3. Podar el árbol cuyo padre es el nodo a eliminar.
4. Redirigir el puntero del nodo padre hacia null para que deje de apuntar al nodo eliminado.

ÁRBOLES ORDENADOS

Surge entonces la necesidad de darle aplicación a los árboles, circunstancia que sólo será posible si se le da cierta propiedad a las relaciones que hay entre los nodos, aquí los árboles ordenados son los que tienen más interés desde el punto de vista de Tipo Abstracto de Dato (TAD), y los que tienen más aplicaciones genéricas. Un árbol ordenado, en general, es aquel a partir del cual se puede obtener una secuencia ordenada siguiendo uno de los recorridos posibles del árbol: InOrden, PreOrden o PostOrden. En estos árboles es importante que la secuencia se mantenga ordenada aunque se añadan o se eliminen nodos.

Existen varios tipos de árboles ordenados algunos de ellos son:

- **Árboles binarios de búsqueda (ABB):** Son árboles de orden 2 que mantienen una secuencia ordenada si se recorren en InOrden.
- **Árboles AVL (Adelson-Velskii Landis):** Son árboles binarios de búsqueda equilibrados, es decir, los niveles de cada rama para cualquier nodo no difieren en más de 1.
- **Árboles perfectamente equilibrados:** Son árboles binarios de búsqueda en los que el número de nodos de cada rama para cualquier nodo no difieren en más de 1. Son por lo tanto árboles AVL también.
- **Árboles 2-3:** Son árboles de orden 3, que contienen dos claves en cada nodo y que están también equilibrados. También generan secuencias ordenadas al recorrerlos en InOrden.
- **Árboles-B:** Caso general de árboles 2-3, que para un orden M, contienen M-1 claves.

CONCLUSIÓN

En base a la lectura y al resumen previamente expuesto se pueden concluir diversos puntos respecto a la estructura de datos árbol.

En primer lugar, es hay que resaltar que un árbol deja de ser una estructura lineal pues cada nodo que lo compone contiene varias relaciones hacia otros nodos.

En segundo lugar, puede notarse que los árboles por sí solos no representan realmente nada, es por ello que es muy importante darle ciertas características o propiedades a las relaciones que hay entre los nodos para que así tomen un significado importante según las necesidades del problema a resolver.

El tercer sitio lo tiene el hecho de que estas características y / o propiedades propician el origen de varios tipos de árboles y que sin embargo, hay algunas operaciones que se efectúan sobre ellos que permanecen iguales o bien muy parecidas. Al mismo tiempo resalta el hecho de que la

naturaleza de cada uno de los tipos de árboles hace que contenga operaciones propias que se aplican únicamente en él y que son parte de las diferentes estrategias que se aplican al manipular datos.

REFERENCIAS

- ***Árbol (Informática)*** Consultado el 14 de abril de 2014 de: es.wikipedia.org/wiki/%C3%81rbol_%28inform%C3%A1tica%29

- ***Capítulo 6: Árboles*** Consultado el 17 de abril de 2014 de: c.conclase.net/edd/index.php?cap=006#inicio

- ***Operaciones Básicas*** Consultado el 17 de Abril de 2014 de: c.conclase.net/edd/index.php?cap=006b#6_3