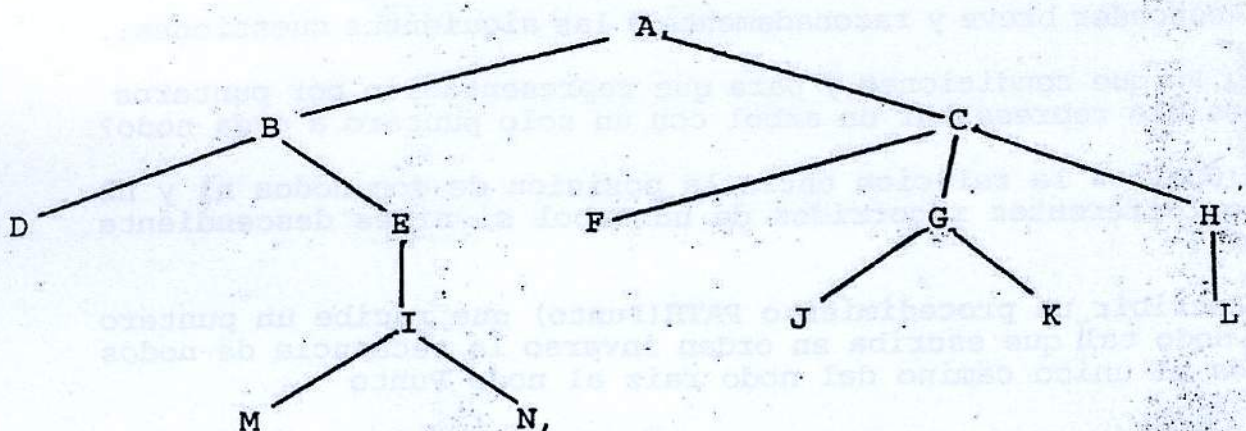


~~ESTRUCTURAS DE DATOS~~

Relacion de Problemas n° 4

1.- Responder las siguientes preguntas sobre el arbol de la figura,

- ¿Que nodos son las hojas?
- ¿Que nodo es la raiz?
- ¿Cual es el padre del nodo C?
- ¿Que nodos son los hijos de C?
- ¿Que nodos son los ancestros de E?
- ¿Que nodos son los descendientes de E?
- ¿Cuales son los hermanos a la derecha de D y E?
- ¿Que nodos est n a la izquierda y a la derecha de G?
- ¿Cual es la profundidad del nodo C?
- ¿Cual es la altura del nodo C?
- ¿Cuantos caminos diferentes de longitud tres hay?
- Listar los nodos del arbol de la figura anterior en preorden, postorden e inorden



2.- Si m y n son dos nodos diferentes del mismo arbol, mostrar que exactamente una de las siguientes afirmaciones es cierta:

- m esta a la izquierda de n
- m esta a la derecha de n
- m es un ancestro propio de n
- m es un descendiente propio de n

3.- Indicar si las condiciones representadas en la fila i columna j de la siguiente tabla pueden ocurrir simultaneamente (Al hablar de ancestros y descendientes nos referimos a ancestros propios)

	$pre(n) < pre(m)$	$in(n) < in(m)$	$post(n) < post(m)$
n es izq. m			
n es dch. m			
n ancestro m			
n descend. m			

4.- Supongamos que tenemos arrays preorder[n], inorder[n] y postorder[n] que contienen la posición preorden, inorden y postorden respectivamente de cada nodo n de un árbol. Describir un algoritmo que nos diga si el nodo i es un ancestro del nodo j, para cada par de nodos i y j. Explicar por que funciona ese algoritmo

5.- Podemos comprobar si un nodo m es un ancestro propio de un nodo n viendo si m precede a n en X-orden pero sigue a n en Y-orden, donde X e Y están entre {pre, in, post}. Determinar todas las parejas X e Y para las cuales es cierto lo anterior

6.- Responder breve y razonadamente a las siguientes cuestiones:

a) ¿En que condiciones y para que representación por punteros es posible representar un árbol con un solo puntero a cada nodo?

b) ¿Cual es la relación entre la posición de dos nodos n1 y n2 en los diferentes recorridos de un árbol si n1 es descendiente de n2?

7.- Escribir un procedimiento PATH(Punto) que recibe un puntero a un nodo tal que escriba en orden inverso la secuencia de nodos que es el único camino del nodo raíz al nodo Punto

8.- Escribir un programa para calcular la altura de un árbol usando cada una de las tres representaciones de árboles conocidas (matricial por listas de hijos y la de hijo-a-la-izquierda hermano-a-la-derecha)

9.- Se define el peso de un árbol como el número de nodos que cuelgan de la raíz del mismo. Diseñar un procedimiento para encontrar subárboles de peso K de un árbol dado

10.- Escribir una función que acepte un puntero a un nodo y retorne el valor verdadero si ese nodo es la raíz de un árbol y falso en caso contrario

11.- Un árbol general difiere de un árbol binario en que puede tener un número variable de subárboles. Sin embargo cualquier árbol general puede ser representado por un árbol binario mediante una ley de correspondencia. Supongamos que t_0, t_1, t_2, \dots son árboles generales y que bt_0, bt_1, bt_2, \dots son sus representaciones binarias. La correspondencia puede establecerse como,

a) Si t_1 es el primer subárbol o el situado más a la izquierda de t_0 , entonces bt_1 es el árbol subbinario de bt_0 ,

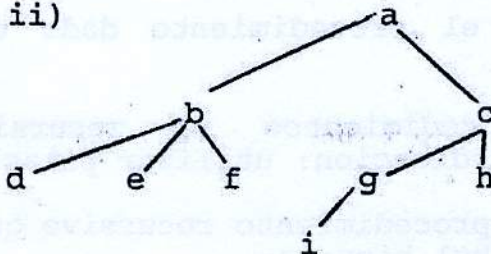
b) Si t_1 y t_2 son subárboles hermanos en este orden, entonces bt_2 es el árbol subbinario derecho de bt_1 ,

Dé los árboles binarios correspondientes a las figuras,

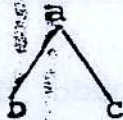
i)



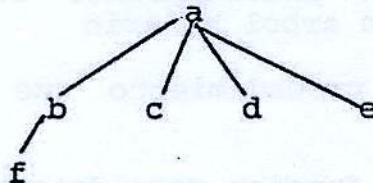
ii)



iii)



iv)



12.- El árbol binario correspondiente a una expresión aritmética se dice que está equilibrado si los resultados de evaluar los símbolos izquierdo y derecho a cada nodo del árbol coinciden.

Diseñar un procedimiento para decidir si un árbol de expresión dado está equilibrado en valoración

NOTA : Diseñar una función $EVALUA(A:Nodo, T:Arbol):integer$ que devuelva el resultado de evaluar el subárbol T cuya raíz es A

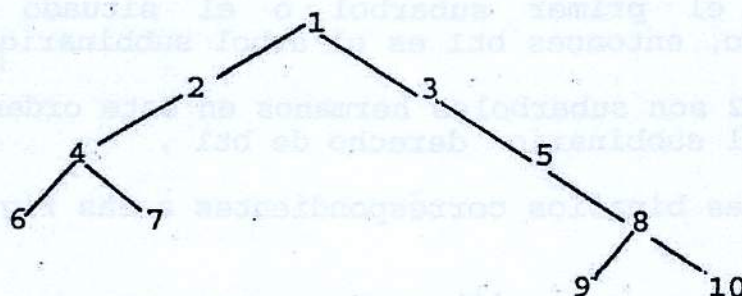
13.- a) Un árbol binario de altura h con el máximo número de nodos se llama árbol binario lleno. Probar que el máximo número de nodos en un árbol binario de altura h es $2^{h+1} - 1$

b) El grado de un nodo es el número de hijos que tiene. Probar que en cualquier árbol binario lleno el número de hojas es uno más que el número de nodos de grado dos

14.- El recorrido en orden por nivel de los nodos de un arbol funciona de la siguiente forma: Primero lista la raiz, luego todos los nodos en la profundidad uno, todos los nodos en profundidad dos, etc. Los nodos de la misma profundidad se listan de izquierda a derecha.

a) Escribir un programa que liste los nodos de un arbol binario en orden por nivel

b) Simular el procedimiento anterior para el siguiente arbol:



c) Modificar el procedimiento dado en a) para arboles cualesquiera

15.- Escribir procedimientos no recursivos para recorrer arbol binario. (Indicacion: utilizar pilas),

16.- Escribir un procedimiento recursivo que cuente el numero de nodos de un arbol binario

17.- Escribir un procedimiento recursivo que encuentre la profundidad de un arbol binario

18.- Escribir un procedimiento que haga un duplicado de un arbol binario

19.- Diseñar una funcion para determinar si un arbol binario es completo

20.- Diseñar una funcion que acepte un puntero a un arbol binario y un puntero a un nodo del arbol, y retorne el nivel del nodo en el arbol

21.- Escribir un procedimiento que acepte un puntero a un arbol y retorne el valor verdadero si el arbol es estrictamente binario y falso en caso contrario

22.- Dada una lista de elementos escalares siempre es posible asociarle un arbol binario de forma que cada nodo del arbol contenga como etiqueta un elemento de la lista, y de forma que el valor de cada nodo N del arbol sea mayor o igual que cualquier valor del subarbol izquierdo de N y menor que cualquier valor del subarbol derecho

a) Comprobar que el recorrido en inorden de ese arbol devuelva la lista inicial ordenada

b) Escribir un programa que ordene listas de escalares utilizando el metodo anterior

23.- Sea un arbol binario cuyas etiquetas son numeros enteros Diremos que el arbol esta parcilmente ordenado si la etiqueta de un nodo es menor que la de sus hijos. Escribir un procedimiento para insertar un nuevo elemento en el arbol, de forma que el nuevo arbol sea binario y parcialmente ordenado

★ 24.- Dos arboles binarios son similares cuando tienen identica estructura de ramificacion, es decir que son ambos vacios o si son no vacios tiene subarboles izqdo. y drcho. similares

Diseñar un procedimiento que compare dos arboles binarios por similaridad

25.- El recorrido en preorden de un arbol binario es ,

1) G E A I B M C L D F K J H,

y en inorden,

2) I A B E G L D C F M K H J,

a) Dibujar el arbol binario,

b) Dar el recorrido en postorden,

c) Diseñar un procedimiento para b) sin usar a) dadas expresiones del tipo 1) y 2),

★ 26.- Para un arbol binario B, podemos construir el arbol binario reflejado B2R 1cambiando los subarboles izqdo. y dcho. en cada nodo excepto la raiz

Diseñar un procedimiento que transforme un arbol binario en el arbol binario reflejado correspondiente

27.- Escribir un programa que lea un numero K de enteros distintos . Ademias, —

a) Genere el arbol binario cuyas etiquetas en los nodos sean los numeros leidos y en el que cada nodo tenga a la izquierda todos sus descendientes menores que el y a la derecha todos sus descendientes mayores que él

b) Recorra el arbol generado en inorden dando como salida las etiquetas

c) Cual seria el arbol generado y su recorrido en inorden si los datos de entrada fueran 128,76,106,402,100,46 —

28.- Se define el peso de un arbol como el numero de nodos que cuelgan de la raiz del mismo. Diseñar un procedimiento para encontrar subarboles de peso K de un ARBOL BINARIO dado

29.- Dado un arbol binario de enteros ordenados, se desea un procedimiento que busque la posicion de un numero entero con un procedimiento recursivo

30.- Diseñar un procedimiento que busque un elemento en un árbol binario de números enteros ordenados, realizado con un proceso iterativo

31.- Escribir una función que acepte un árbol binario y dos nodos que no son raíces en el árbol y retorne el ancestro común más profundo

32.- Un árbol BSP es un árbol binario parcialmente ordenado que cumple la condición de que sus hojas están ordenadas de izquierda a derecha de menor a mayor. Diseñar un procedimiento para insertar un nodo en un árbol de este tipo de forma que tras la inserción siga cumpliendo las condiciones de definición

33.- a) Convertir la expresión $((a+b)+c*(d+e)+f)*(g+h)$ a expresión prefija y postfija

b) Dibujar las representaciones en árbol para las expresiones prefijas:

* a + b * c + d e.

* a + * b + c d e.

¿Cuáles son las correspondientes expresiones infijas parentizadas?

34.- Dada una expresión postfija generar el árbol asociado de forma que el recorrido en postorden del mismo devuelva la expresión original

35.- Sean las expresiones aritméticas enteras con las operaciones de suma y producto representadas mediante árboles binarios con etiquetas de tipo string. Se supone que existe una variable global (valores) de tipo vector, cuyos elementos son registros y que contienen los posibles identificadores de las variables que aparecen en las expresiones y sus valores asociados. El número de identificadores válidos está almacenado en una variable global (n)

a) Contruir una función VALOR(l:string):integer, que calcule el valor asignado a un identificador l

b) Escribir un procedimiento que calcule el valor de una expresión dada, supuesta esta válida

36.- Dibujar el árbol de expresión correspondiente a,

$(A - (B - C) * (D + E / (F - G) * H) * (I + J) - K) / L$

37.- Implementar las rutinas primitivas de los árboles, para la implementación de:

a) Por matrices,

b) Lista de hijos


```

Lista ListarAvlClaves(ArbolAvl a)
{
    Lista l;

    l = CrearLista(((struct avlCab *)a)→tamanoClave);
    ListarAvlClavesCeldas(((struct avlCab *)a)→raiz, PrimeroLista(l), l);

    return l;
}

```

Como se puede observar es distinta a la ofrecida en la sección de ejercicios resueltos sólo en lo que respecta al orden de dos líneas. ¿Esta solución es correcta? ¿Si no lo es, por qué no funciona?

9. Implementar en un fichero *avlsort.c* una función similar a la especificada en la sección de ejercicios resueltos como *heapsort* pero usando en lugar de un APO un árbol AVL y aprovechando el listado en inorden de la función *ListarAvlClaves* de manera que ordene un array de n elementos. Dichos elementos serán de un tipo genérico.
10. Construir cada uno de los B-árboles que se van generando conforme se van insertando los números 1, 9, 32, 3, 53, 43, 44, 57, 67, 7, 45, 34, 23, 12, 23, 56, 73, 65, 49, 85, 89, 64, 54, 75, 77, 49 en un B-árbol de orden 5.
11. Supongamos se insertan un conjunto de elementos en un B-árbol en un determinado orden. ¿La altura del B-árbol resultado es independiente del orden en que se han insertado los elementos?. Razónese la respuesta.
12. Dos árboles binarios T_1 y T_2 se dicen isomórficos si T_1 puede transformarse en T_2 intercambiando los hijos izquierdo y derecho de algunos nodos. Diseñar un algoritmo para comprobar si 2 árboles binarios son isomórficos.
13. Al aplicar el método de ordenación Heapsort se aplica un procedimiento que podemos denominar "criba", mediante el cual los elementos se organizan en forma de montículos (heap): sucesión de elementos x_1, \dots, x_n / $x_i \leq x_{2i}$ $i = 1, \dots, \frac{n}{2}$
 Para ordenar los elementos a partir del montículo x_1, \dots, x_n se repite el proceso de intercambiar el primero con el último y aplicar el proceso de criba a los $n - 1$ primeros elementos. ¿Podría el Heapsort implementarse aplicando el algoritmo de criba a x_2, \dots, x_n , x_3, \dots, x_n , x_4, \dots, x_n, \dots en lugar de hacerlo a través del proceso anteriormente expuesto?. Razonar la respuesta en términos de eficiencia.
14. Se define el peso de un nodo en un árbol binario A como el número de nodos que son descendientes suyos. ¿Cómo construirías un algoritmo eficiente para determinar si dos subárboles concretos en A tienen el mismo peso?
15. a.- ¿Puede construirse de forma única un árbol binario dado conociendo su preorden y el peso de cada nodo (el número de nodos que son descendientes suyos exceptuándose él mismo)?

b.- ¿Puede reconstruirse de forma única un ABB de forma única dado su inorden? ¿Y dados el preorden y el postorden?

16. Construir un ABB y un APO con las claves 50, 25, 75, 10, 40, 60, 90, 35, 45, 70, 42.

17. Se define la frontera de un árbol binario como la sucesión de sus hojas escrita en orden de izquierda a derecha. Dados 2 árboles binarios A y B con fronteras respectivas $f(A)$ y $f(B)$, ¿cómo construirías un algoritmo eficiente para determinar si $f(A) = f(B)$?

18. Construir un ABB equilibrado a partir de las claves 10, 75, 34, 22, 64, 53, 41, 5, 25, 74, 20, 15, 90.

19. ¿Bajo qué condiciones puede un árbol ser parcialmente ordenado y binario de búsqueda simultáneamente? Razona la respuesta.

20. Sean A y B dos árboles binarios distintos. ¿Puede ocurrir simultáneamente: $\text{Pre}(A) = \text{Post}(B)$ y $\text{Post}(A) = \text{Pre}(B)$?

21. Se define la densidad de un árbol binario A como el número de hojas multiplicado por la profundidad del árbol. Construir un algoritmo eficiente para determinar si 2 árboles binarios diferentes tienen la misma densidad.

22. Construir un ABB equilibrado mediante la inserción (en orden) de las claves 75, 22, 34, 64, 41, 53, 5, 25.

23. ¿Puede reconstruirse un ABB de forma unívoca dado su recorrido en inorden? ¿Y dado su recorrido en preorden?

24. Construir un ABB equilibrado con las claves 10, 75, 34, 22, 64, 53, 41, 5, 25

25. ¿Puede reconstruirse un APO de forma unívoca dado su recorrido en preorden?

26. Construir un ABB equilibrado (AVL) con las claves 20, 45, 14, 22, 24, 13, 43, 2, 25 insertadas en ese orden.

27. Construir el TDA APO a partir del TDA Pila

28. Un "Montón max-min" es una estructura que permite realizar las operaciones *eliminar-maximo* y *eliminar-minimo* en un tiempo $O(\log_2(n))$, y que tiene como propiedad fundamental el que para cualquier nodo X a profundidad par la clave almacenada en X es mayor que la del padre pero menor que la del abuelo (cuando existen), y para cualquier nodo X a profundidad impar, la clave almacenada en X es menor que la del padre pero mayor que la del abuelo (cuando existen). Poner un ejemplo de "Montón max-min" con más de 10 claves.

29. Un "Montón min-max" es una estructura que permite realizar las operaciones *eliminar-minimo* y *eliminar-maximo* en un tiempo $O(\log_2(n))$, y que tiene como propiedad fundamental el que para cualquier nodo X a profundidad par la clave almacenada en X es menor que la del padre pero mayor que la del abuelo (cuando existen), y para cualquier nodo X a profundidad impar, la

clave almacenada en X es mayor que la del padre pero menor que la del abuelo (cuando existen).

- a) Poner un ejemplo de "Montón min-max" con más de 10 claves.
- b) Diseñar un algoritmo para insertar un nuevo nodo en la estructura

30. Se define el volumen de un árbol binario A como el número de hojas multiplicado por la altura del árbol. Construir un algoritmo eficiente para determinar si 2 árboles binarios diferentes tienen el mismo volumen.
31. Construir un ABB equilibrado mediante la inserción (en orden) de las claves 10, 75, 22, 34, 64, 41, 53, 5, 25.

32. Sea A un árbol en el que cada nodo que no es hoja tiene 2 hijos. Escribir un procedimiento para convertir un listado en preorden de A en un listado en postorden.

Se define un P-árbol como un árbol binario parcialmente ordenado que además cumple la condición de que para cada nodo N, su hijo a la derecha tiene un valor intermedio entre él y su hijo a la izquierda, es decir, $\text{etiqueta}(\text{hizqda}(N,A),A) > \text{etiqueta}(N,A) > \text{etiqueta}(\text{hdrcha}(N,A),A)$. Diseñar un procedimiento de inserción de un nuevo nodo.

33. Insertar los enteros 7,2,9,0,5,6,8,1,10 en un árbol binario de búsqueda y en un APO aplicando reiteradamente el procedimiento de inserción. Obtener los árboles resultantes de aplicar el borrado de 7 y de 2.
34. Usualmente en un árbol binario de búsqueda (ABB), los nuevos nodos se insertan en las hojas, pero también es posible hacer crecer un ABB por la raíz, usando una clave de búsqueda K para partir el ABB en 3 componentes: un ABB izquierdo conteniendo todos los nodos con claves menores que K, el propio nodo K y un ABB derecho conteniendo todos los nodos con claves mayores que K. Diseñar un procedimiento para insertar nuevos nodos en un ABB construido de esta manera.

Capítulo

Tabla

7.1 Inti

Una aproximación, no por cierto directa, a la respuesta es,

La primera respuesta es, que tal función de tabla de tamaño $40^{30} = 1.15 \cdot 10^{61}$ de cada 10 millones de factible sólo es a priori. Existe la posibilidad de organizar las palabras clave.

Las funciones incluso para si en una red de ellas haya una función aleatoria de claves no caídas.

En consecuencia, tienen la posibilidad de (k_i, k_j) posible de caer el otro será.

7.2 Fu

El primer conjunto de claves en los

Relación de Problemas sobre árboles

1. Escribir una función para calcular la altura de un árbol cualquiera.
2. Escribir una función no recursiva para calcular la altura de un árbol cualquiera.
3. Indicar una función no recursiva para recorrer un árbol general en postorden.
4. Indicar e implementar un algoritmo no recursivo para realizar un recorrido en preorden sobre un árbol general.
5. Indicar e implementar un algoritmo no recursivo para realizar un recorrido en inorden sobre un árbol general.
6. Implementar una función no recursiva que imprima el recorrido en postorden de un árbol general.
7. Escribir una función que acepte un valor de tipo *Nodo* y un árbol general *T* y devuelva el nivel del nodo en el árbol. Considérese que la raíz está a nivel 0.
8. ¿Puede determinarse un árbol general a partir de los recorridos en inorden y postorden?. Si la respuesta es negativa poner un contraejemplo.
9. Resuelva las dos cuestiones siguientes:
 - (a) Un árbol binario de altura h con el máximo número de nodos se llama árbol binario completo. Probar que el máximo número de nodos en un árbol binario de altura h es $2^{h+1} - 1$.
 - (b) El grado de un nodo es el número de hijos que tiene. Probar que en cualquier árbol binario completo el número de hojas es uno más que el número de nodos de grado 2.
10. Escribir una función no recursiva para recorrer un árbol binario en preorden.
11. Implementar una función no recursiva para recorrer un árbol binario en inorden.
12. Implementar una función no recursiva para recorrer un árbol binario en postorden.
13. Escribir una función recursiva que encuentre el número de nodos de un árbol binario.
14. Escribir una función recursiva que encuentre la altura de un árbol binario.
15. Escribir una función que realice la reflexión de un árbol binario.
16. Supongamos que tenemos una función *valor* tal que dado un valor de tipo *char* (una letra del alfabeto) devuelve un valor entero asociado a dicho identificador. Supongamos también la existencia de un árbol de expresión *T* cuyos nodos hoja son letras del alfabeto y cuyos nodos interiores son los caracteres *, +, -, /. Diseñar una función que tome como parámetros un nodo y un árbol binario y devuelva el resultado entero de la evaluación de la expresión representada.
17. El recorrido en preorden de un determinado árbol binario es:
G E A I B M C L D F K J H
y en inorden
I A B E G L D C F M K H J
Resolver:
 - (a) Dibujar el árbol binario.
 - (b) Dar el recorrido en postorden.
 - (c) Diseñar una función para dar el recorrido en postorden dados los recorridos en preorden e inorden y escribir un programa para comprobar el resultado del apartado anterior.
18. Utilizando una estructura de tipo árbol parcialmente ordenado en la cual la inserción y extracción de elementos toman un tiempo $O(\log n)$ se puede construir un algoritmo de ordenación en tiempo $O(n \log n)$ dado que los elementos pueden ser extraídos de forma ordenada. El algoritmo¹ consistiría en dos pasos bien diferenciados:

¹ Este algoritmo recibe el nombre de *heapsort* dado que utiliza un "heap" para ordenar