

Práctico 5: Árboles

(Finalización 28/09)

Licenciatura en Ciencias de la Computación - Profesorado en Ciencias de la Computación

Ejercicio 1:

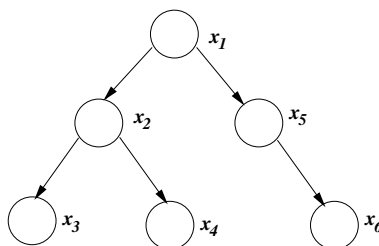
Dada la secuencia de números:

4, 19, -7, 49, 100, 0, 22, 12, -12

- a) Construir un árbol binario de búsqueda, insertando la secuencia de elementos dada.
- b) Insertar el elemento 10 en el árbol.
- c) Eliminar el elemento 19 del árbol (utilizar la política: mayor de los menores).
- d) Insertar el elemento 1 en el árbol.

Ejercicio 2:

¿Cuántas secuencias de entrada distintas, de todas las posibles secuencias, generarían el siguiente árbol? Mostrar tres de ellas.



Ejercicio 3:

Dada una relación y sabiendo que se quieren resolver servicios asociativos sobre la misma, se pide: graficar el árbol binario de búsqueda (ABB) que se obtiene, en cada caso, al incorporar las nuplas de acuerdo a las secuencias dadas:

- i) $R \subseteq \text{Código-Postal} \times \text{Ciudad}$ siendo el asociante *Código-Postal*. Las nuplas son: (4700, Catamarca), (3600, Formosa), (5500, Mendoza), (8300, Neuquén), (3400, Corrientes), (5400, San Juan), (4600, Jujuy), (3000, Santa Fe), (4000, Tucumán), (4200, Santiago del Estero), (9410, Ushuaia), (8500, Viedma).
- ii) $S \subseteq \text{Patente} \times \text{Marca}$ siendo el asociante *Patente*. Las nuplas son: (FUP980, Citroën), (EAX720, Alfa Romeo), (BGG675, Renault Scenic), (CDG890, Volkswagen), (AZT215, Chevrolet Zafira), (BGH287, Toyota), (PYU888, Chrysler), (CDG920, Ford), (PHI789, Peugeot 5004), (MWT567, Chevrolet Astra), (PNY367, Audi), (NRT378, Fiat).

Ejercicio 4:

Para uno de los árboles obtenidos en el ejercicio anterior, eliminar los elementos en el mismo orden en el que fueron incorporados. Dejar en claro la política de reemplazo que utiliza.

Ejercicio 5:

Dar las secuencias de nodos que se obtendrían si los siguientes árboles binarios fueran visitados usando:

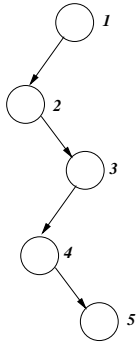
a) barrido pre-orden

b) barrido post-orden

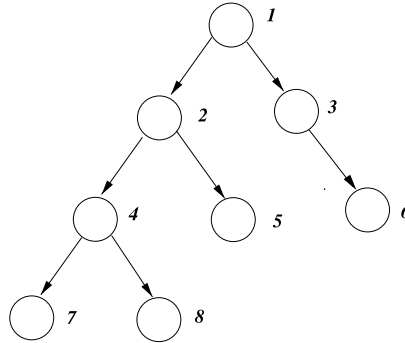
c) barrido en orden

d) barrido por niveles

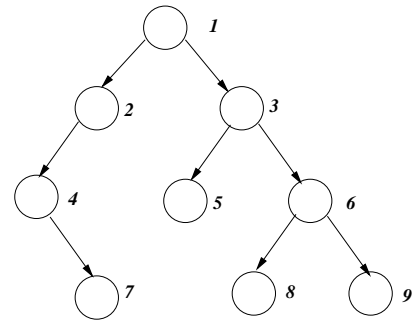
1)



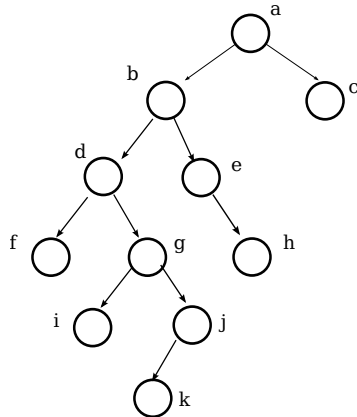
2)



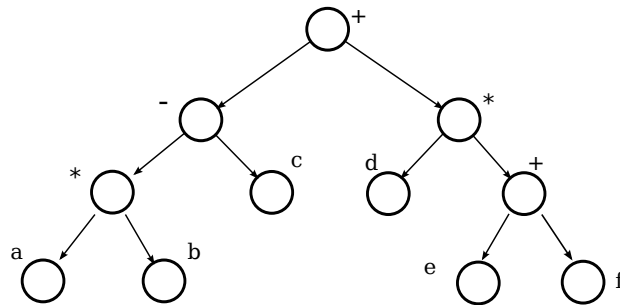
3)



4)



5)



Ejercicio 6:

Desarrollar los algoritmos *iterativos* para los distintos barridos de un árbol binario de búsqueda.

Ejercicio 7:

Sabiendo que el recorrido en *preorden* de un determinado árbol binario es: **GEAIBMCLDFKJH** y el recorrido en *inorden* es: **IABEGLDCFMKHJ**.

a) Dibujar el árbol binario.

b) Dar el recorrido en postorden.

Ejercicio 8:

Considerando una relación $R \subseteq X \times Y$ almacenada en un árbol binario de búsqueda (ABB), se pide:

a) Desarrollar los operadores *LOCALIZACIÓN*, *EVOCACIÓN*, *ALTA* y *BAJA*.

b) Desarrollar las evocaciones extremas NO destructivas: *Recuperar MÍNIMO* y *Recuperar MÁXIMO*.

Ejercicio 9:

Para cada uno de los árboles obtenidos en el **Ejercicio 3**, obtener el esfuerzo medio y máximo de localización exitosa y no exitosa **a posteriori**. Agregue las hipótesis que considere necesarias.

Ejercicio 10:

Diseñar la función:

```
int nodosCaminoMin(a:arbol)
```

que dado un árbol binario no vacío determine y devuelva el número de nodos existentes en el camino más corto desde la raíz a una hoja. ¿Cuál sería, en notación asintótica, el costo máximo a priori de este algoritmo, si se considera como función de costo: celdas consultadas?

Ejercicio 11:

Implementar una función booleana que, dados dos árboles binarios con igual cantidad y disposición de nodos, responda *True* si el primero es menor al segundo y *False* en otro caso. Diremos que un árbol binario es menor a otro, si los elementos del primero, en los nodos coincidentes en posición, son menores que los del segundo.

Ejercicio 12:

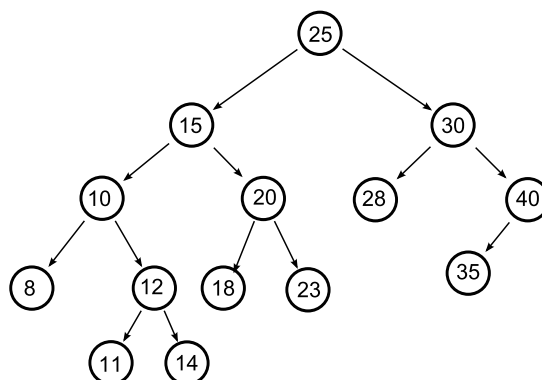
Sea X un conjunto de números reales, se define el *rango* del conjunto X como la diferencia entre el mayor y el menor valor de X . Suponiendo que el conjunto X se encuentra almacenado en un ABB, se pide:

- Desarrollar en pseudo-código un algoritmo que permita calcular el rango de X .
- Obtener el esfuerzo máximo a priori, en notación asintótica, de realizar dicha operación considerando el algoritmo desarrollado por Ud. en a). Agregue las hipótesis que considere necesarias.

Ejercicio 13:

Sea el conjunto $X \subseteq \mathbb{N}$ almacenado en un ABB se pide:

- Desarrollar, en pseudocódigo, una rutina que devuelva los elementos de X almacenados en el ABB que caigan en el intervalo $[a, b]$.
- Ejecute la rutina desarrollada en a) para los intervalos $[12, 30]$ y $[8, 20]$ sobre el siguiente ABB.



Ejercicio 14:

Dados N elementos de un conjunto almacenado en una *LSD*, se pide desarrollar un algoritmo que realice su ordenamiento, de *mayor a menor*, utilizando como estructura auxiliar un *ABB* (*Treesort*). Suponga para ello que cuenta con todos los operadores básicos sobre las estructuras ya programados.

Ejercicio 15:

a) Dar el árbol de expresión para cada una de las siguientes expresiones:

i) $n - (\log y + \log x)$

ii) $b + (b^2 - 4 * a * c) / 2 * a$

iii) $(\neg(p_1 \vee p_3) \Rightarrow (p_2 \wedge \neg p_2))$

b) Usando cada uno de los árboles obtenidos en a) obtener la notación prefijo y posfijo de cada una de las expresiones anteriores.

Ejercicio 16:

Realizar las gráficas de los árboles balanceados en altura (*AVL*) que se obtienen al incorporar las nuplas de las relaciones R y S del **Ejercicio 3**. Seguir las secuencias dadas en el mismo, y graficar cada paso mostrando las rotaciones realizadas cuando corresponda.

Ejercicio 17:

Eliminar los elementos de uno de los árboles del ejercicio anterior, utilizando:

a) orden FIFO (first in first out)

b) orden LIFO (last in first out)

Graficar paso a paso este proceso. Utilizar como política de reemplazo el mayor de los menores.

Ejercicio 18:

Suponiendo que $R \subseteq X \times Y$ está almacenada en un *AVL*, se pide:

a) Analizar los casos de *ALTA* y *BAJA* que pueden presentarse, dando un ejemplo para cada uno de ellos.

b) Desarrollar los operadores *ALTA* y *BAJA*.

Ejercicios Adicionales

Ejercicio 1:

Demostrar que todo árbol ternario tiene un número *impar* de hojas.

Ejercicio 2:

Sea A un árbol ternario, se pide:

- a) Definir recursivamente $i(A)$ y $e(A)$.
- b) Definir recursivamente $I(A)$ y $E(A)$.
- c) Encontrar y demostrar la relación que existe entre $i(A)$ y $e(A)$.
- d) Encontrar y demostrar la relación que existe entre $I(A)$ y $E(A)$.