

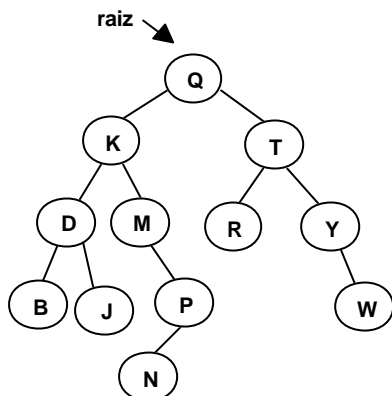
Programación Orientada a Objetos
Analista de Sistemas – Licenciatura en Sistemas
Trabajo Práctico Nº 07: Árboles

1. ¿Cuál es el número total máximo de nodos que tiene un árbol binario de N niveles?

- $N^2 - 1$
- $2^{N+1} - 1$
- 2^N
- 2^{N+1}

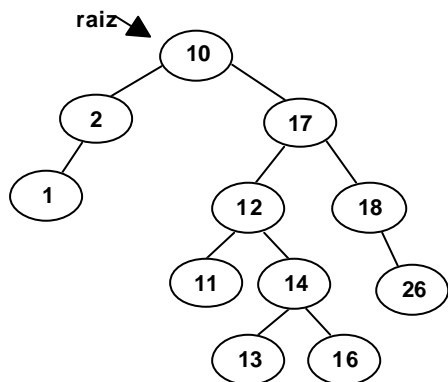
i.

2. Dado el siguiente árbol binario:



- ¿Cuáles son los antecesores del nodo P?
- ¿Cuáles son los descendientes del nodo K?
- ¿Cuál es el máximo número posible de nodos del árbol en el nivel del nodo W?
- ¿Cuál es el máximo número posible de nodos del árbol en el nivel del nodo N?
- ¿Cuál es la altura del árbol?
- ¿Qué nodos se hallan en el nivel 2?
- ¿Qué grado tienen los nodos R, K y M?

3. Dado el siguiente árbol binario:



- ¿Cuáles son los antecesores del nodo 13?
- ¿Cuáles son los descendientes del nodo 17?
- ¿Cuál es la altura del árbol?
- ¿Qué valores están en los nodos hojas?
- ¿Qué valores están en los nodos interiores?

Para los ejercicios 4 a 8, hacerlos a mano y luego corroborar los resultados en máquina.

4. El árbol binario del ejercicio 2 es de búsqueda, graficar el árbol resultante de efectuar las siguientes operaciones:

a. Añadir nodo C.	b. Suprimir nodo M.
c. Añadir nodo Z.	d. Suprimir nodo Q.
e. Añadir nodo X.	f. Suprimir nodo R.

5. El árbol binario del ejercicio 3 es de búsqueda, graficar el árbol resultante de efectuar las siguientes operaciones:

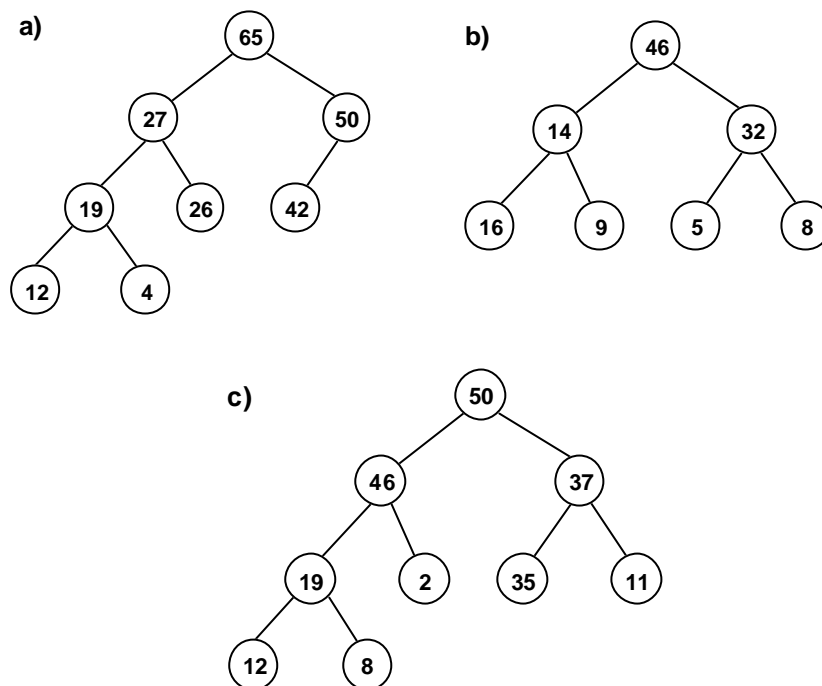
a. Añadir nodo 3.	b. Suprimir nodo 18.
-------------------	----------------------

Trabajo Práctico Nº 07: Arboles

c. Añadir nodo 30.	d. Suprimir nodo 11.
e. Añadir nodo 25.	f. Suprimir nodo 17.

6. Para los árboles binarios de los ejercicios 2 y 3 mostrar qué se imprimiría con lo siguiente:
 - a. Un recorrido en inorden del árbol.
 - b. Un recorrido en postorden del árbol.
 - c. Un recorrido en preorden del árbol.
7. En un árbol binario de búsqueda de enteros:
 - a. Insertar los enteros 7,2,9,0,5,6,8,1
 - b. Mostrar el resultado obtenido al suprimir el nodo 7 y después el nodo 2 del árbol anterior.
8. El atributo *info* de los nodos en un árbol binario de búsqueda contiene cadenas cortas de caracteres.
 - a. Mostrar cómo quedaría un árbol, después de insertar las siguientes palabras (en el orden indicado):
mono canario burro ciervo
yak morsa buitre pingüino perdiz
 - b. Mostrar cómo quedaría el árbol, si las mismas palabras se insertaran en este orden:
perdiz morsa burro ciervo mono
buitre yak pingüino cebra canario
 - c. Mostrar cómo quedaría el árbol, si las mismas palabras se insertaran en este orden:
cebra yak morsa buitre perdiz
pingüino mono burro ciervo canario
9. Implementar el algoritmo buscar de árboles binarios de búsqueda en forma recursiva.
10. Implementar un iterador de árbol binario de búsqueda.
11. Agregar a la clase Arbol métodos para:
 - a. Determinar la altura del árbol.
 - b. Determinar la cantidad de elementos del árbol.
 - c. Imprimir las hojas del árbol.
 - d. Imprimir los elementos del nivel n.
 - e. Mostrar los descendientes de un elemento.
 - f. Mostrar los ascendientes de un elemento.
 - g. Buscar un valor y mostrar el padre.
 - h. Buscar un valor mostrar su hermano.
12. Un árbol binario contiene valores enteros en el atributo *info* de cada nodo. Escribir un método SUMACUADRADO que devuelva la suma de los cuadrados de los valores del árbol.
13. Escribir un método iterativo Antecesor que imprima los antecesores de un nodo cuyo atributo *info* contiene un valor Num. Num solo aparece una vez en el árbol. No imprimir Num.
14. Escriba un programa que permita ingresar, eliminar, buscar y recorrer un árbol binario de búsqueda, en el cual se pueden dar varias ocurrencias de un mismo valor de información, sin que esto ocasione duplicación de nodos en memoria. Además el programa deberá informar a pedido del usuario:
 - a. Cantidad total de nodos en árbol.
 - b. Profundidad del árbol.
 - c. Valores de un determinado nivel.
 - d. Antecesores de un nodo.
15. Generar dos árboles binarios de búsqueda en forma random y realizar las siguientes operaciones:
 - a) mostrar la unión de ambos conjuntos.
 - b) mostrar la intersección.
 - c) mostrar la diferencia.
16. Generar un array de punteros arboles binarios de búsqueda (en forma random).
 - a) implementar la búsqueda de un elemento de manera tal, que se informe en que árboles dicho elemento se encuentra.
 - b) almacenar todos los árboles en único conjunto.

17. Decir si los siguientes árboles son completos, llenos, montículos o ninguno de los anteriores:

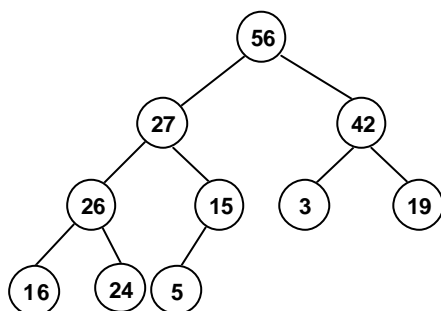


18. Una cola de prioridad que contiene caracteres se implementa como un montículo, la precondition afirma que esta cola de prioridad no puede contener elementos duplicados. ¿Qué valores pueden almacenarse en las posiciones 8 a 10 del array, de forma que las prioridades de un montículo se satisfagan?

Montículo

Z	F	J	E	B	G	H	?	?	?
[1]	[2]	[3]	[4]	[]	[6]	[7]	[8]	[9]	[10]

19. Una cola de prioridad se implementa como un montículo:



Mostrar como quedaría el montículo anterior después de esta serie de operaciones:

```
ColaDePrioridad CP = new ColaDePrioridad();
int X, Y, Z;
```

```
CP.insertar(28);
CP.insertar(2);
CP.insertar(40);
X = CP.suprimir();
Y = CP.suprimir();
Z = CP.suprimir();
```

b. ¿Cuáles serían los valores de X, Y y Z después de la serie de operaciones del inciso a?

Programación Orientada a Objetos
Analista de Sistemas – Licenciatura en Sistemas
Trabajo Práctico Nº 07: Árboles

20. Agregar a la clase Montículo los recorridos inorden y postorden.
21. Escribir los métodos limpiarColaPrioridad() y estaLlena(), usando la implementación de montículo.
22. Una cola de Prioridad se implementa como una lista enlazada. Escribir la clase para esta implementación.(insertar, suprimir, etc.)
Comparar con las operaciones con las correspondientes a la implementación de montículo. ¿Qué implementación es mejor/peor para cada operación?
- Nota: ordenar la lista de mayor a menor.
23. Una Cola de prioridad se implementa como un árbol binario de búsqueda. Escribir la clase para esta implementación.(insertar, suprimir, etc.)
Comparar con las operaciones con las correspondientes a la implementación de montículo. ¿Qué implementación es mejor/peor para cada operación?
24. Una cola de Prioridad se implementa como un montículo, está ordenada para acceder al elemento más pequeño antes que al elemento mayor. ¿Cómo necesitarían ser modificados los algoritmos en la Cola de prioridad? ¿Cómo necesitarían ser modificados los algoritmos en las operaciones sobre montículos? ¿Se puede implementar un montículo en una lista enlazada?
25. Realizar los siguientes ejercicios utilizando una cola de prioridad implementada en un montículo.

Los usuarios del sistema tienen prioridades relativas de acuerdo con su número de ID:

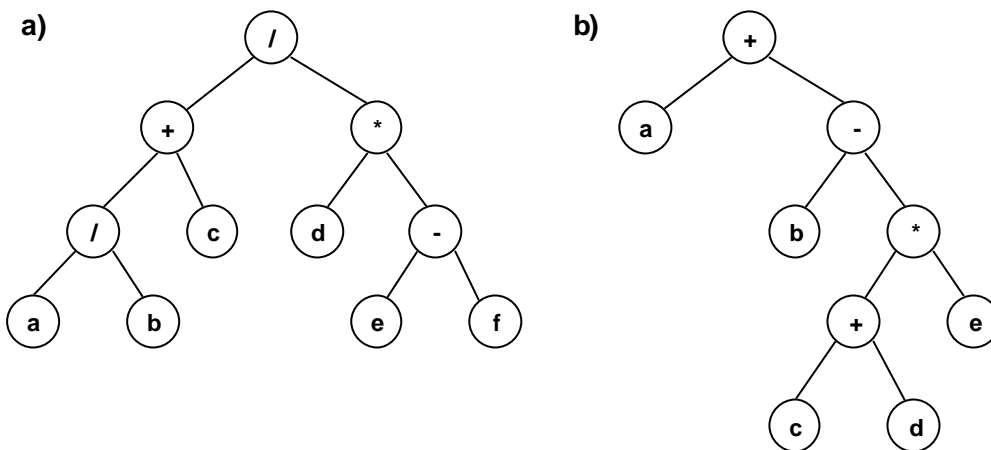
Usuarios 0 – 99	mayor (por ejemplo, ejecutivos de la empresa)
Usuarios 100 – 199	siguiente más alta (por ejemplo, secretarios ejecutivos)
Usuarios 200 – 299	siguiente más alta (por ejemplo, jefes de grupo)
--	
--	
--	
Usuarios 800 – 899	anterior a la más baja (por ejemplo, programadores)
Usuarios 900 – 999	más baja (sus trabajos sólo se ejecutarán a las 3 de la madrugada los fines de semana)

Escribir un método **añadirTrabajo** que reciba un número de **id** de usuario y un **token** representando el trabajo que hay que ejecutar y lo añada a la cola.

Escribir un método **obtenerSigTrabajo** que devuelva el **token** del trabajo para ejecutar.

Hay que apagar el sistema para el mantenimiento. Todos los trabajos que estén esperando para ejecutarse se quitarán de la cola. Sin embargo, este es un sistema amistoso que notifica a los usuarios

26. Escribir un programa que evalúe una expresión aritmética. La expresión puede ingresarse en la notación que se crea más adecuada.
27. Mostrar la notación en preorden, inorden y postorden de los siguientes árboles binarios de expresión (Recordar que debe utilizar paréntesis para la notación en inorden).



28. Evaluar el árbol binario de expresión del ejercicio 11.a con los siguientes valores para las variables a - f:
a = 6; b = 2; c = 5; d = 2; e = 8; f = 7
29. Evaluar el árbol binario de expresión del ejercicio 11.b con los siguientes valores para las variables a - e:
a = 5; b = 12; c = 2; d = 3; e = 2
30. Dibujar el árbol binario de una expresión que representa la siguiente expresión en preorden:
+ * a b / + c d e
31. Dibujar el árbol binario de una expresión que representa la siguiente expresión en inorden:
(((a - b) + c) * ((d + e) / f)) - g
32. Dibujar el árbol binario de una expresión que representa la siguiente expresión en postorden:
a b + c / d *
33. Escribir un método recursivo para imprimir la representación en postorden de un árbol binario de expresión. Usar la implementación con registro variante de los nodos del árbol.