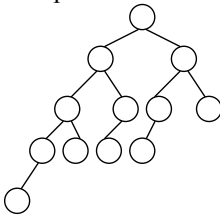
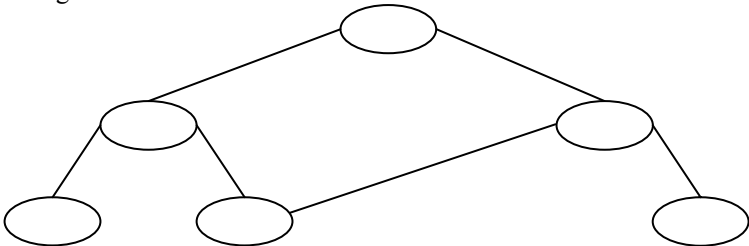


DNI:

Modalidad 0

	V	F		
El siguiente árbol está balanceado con respecto a la altura 	<input type="checkbox"/>	<input type="checkbox"/>	1.	V
La siguiente función de C++, <code>int& Incremento(int valor){valor=valor+5;return valor};</code> , devuelve el resultado por referencia.	<input type="checkbox"/>	<input type="checkbox"/>	2.	V
En las colas, las inserciones y borrados se realizan por el mismo extremo.	<input type="checkbox"/>	<input type="checkbox"/>	3.	F
La siguiente estructura es un árbol binario: 	<input type="checkbox"/>	<input type="checkbox"/>	4.	F
Un árbol completo es un árbol completamente equilibrado	<input type="checkbox"/>	<input type="checkbox"/>	5.	F
Los árboles AVL son árboles balanceados con respecto a la altura de los subárboles.	<input type="checkbox"/>	<input type="checkbox"/>	6.	V
En un árbol 2-3, la diferencia en número de nodos entre los subárboles de la raíz es como mucho 1.	<input type="checkbox"/>	<input type="checkbox"/>	7.	F
Un árbol rojo-negro, en el que no hay ningún enlace rojo, es un árbol binario completo.	<input type="checkbox"/>	<input type="checkbox"/>	8.	F
Un árbol binario de búsqueda con altura 7 y 127 nodos es un árbol B con m=2	<input type="checkbox"/>	<input type="checkbox"/>	9.	V
En un árbol m-camino de búsqueda, todos los nodos excepto la raíz tienen al menos m/2 hijos.	<input type="checkbox"/>	<input type="checkbox"/>	10.	F
En la dispersión cerrada se pueden producir colisiones entre claves no sinónimas	<input type="checkbox"/>	<input type="checkbox"/>	11.	V
En un montículo doble, un elemento “j” del montículo máximo es el simétrico de un único elemento “i” del montículo mínimo.	<input type="checkbox"/>	<input type="checkbox"/>	12.	F
Un árbol Rojo-Negro cumple las propiedades de un árbol Leftist.	<input type="checkbox"/>	<input type="checkbox"/>	13.	F
Al representar un grafo no dirigido con una matriz de adyacencia, su diagonal principal (casillas i,i) siempre tendrá valores Falso.	<input type="checkbox"/>	<input type="checkbox"/>	14.	V

Examen TAD/PED septiembre 2005

Normas: ♦ Tiempo para efectuar el ejercicio: **2 horas**

- En la cabecera de cada hoja **Y EN ESTE ORDEN** hay que poner: **Apellidos, Nombre. LAS PREGUNTAS 1 Y 2 SE ENTREGARÁN JUNTAS, IGUALMENTE LA 3 y 4.**
- Se dispone de 20 minutos para abandonar el examen sin que corra convocatoria.
- Las soluciones al examen se dejarán en el campus virtual.
- Se puede escribir el examen con lápiz, siempre que sea legible
- Todas las preguntas tienen el mismo valor.** Este examen vale el 60% de la nota de teoría.
- Publicación de notas de exámenes:** 13 de septiembre. **Fecha de revisión de exámenes:** 19 septiembre, de 11:30 a 13:00 h. en el laboratorio OIN01 de la EPSA IV.

♦ Los alumnos que estén en 5ª o 6ª convocatoria deben indicarlo en la cabecera de todas las hojas

1. Completa en esta misma hoja las ecuaciones que aparecen a continuación y que expresan el comportamiento de las operaciones de: a) *obtener* en una lista de acceso por posición, b) *resta* en el conjunto de los números Naturales en el que sólo existen las operaciones *cero*: \rightarrow *natural* y la operación *suc*: *natural* \rightarrow *natural* (devuelve el sucesor de un número Natural). Se asume que el primer operando de la *resta* es siempre mayor o igual que el segundo.

a) *obtener*(lista,posicion) \rightarrow item

obtener(crear(),p) =

si p = primera (inscabeza(l₁,x)) entonces *obtener*(inscabeza(l₁,x),p) =

sino *obtener*(inscabeza(l₁,x),p) =

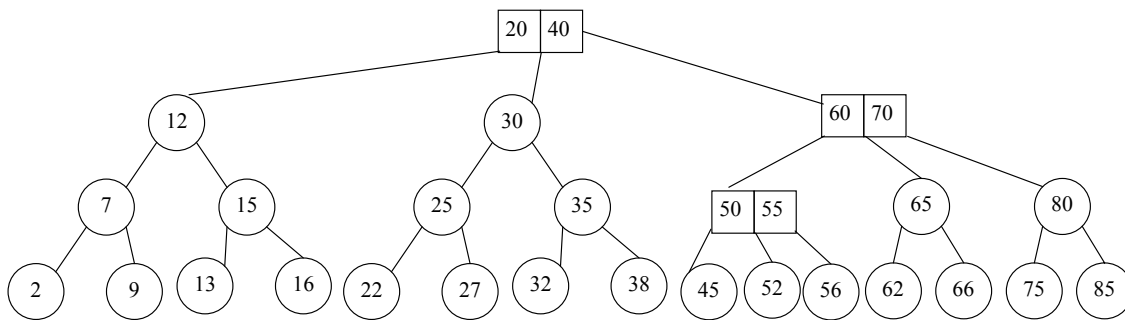
b) *resta*(natural,natural) \rightarrow natural

resta(,) =

resta(,) =

Donde: l₁ ∈ lista x ∈ item p ∈ posicion a, b ∈ natural

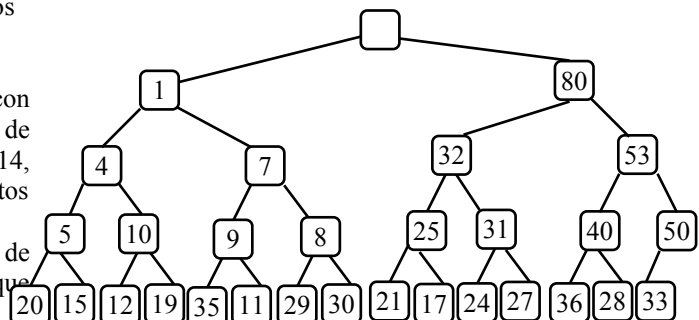
2. Dado el siguiente árbol 2-3 borrar los elementos 40, 30 y 12. (Criterios: (1) si el nodo tiene dos hijos hay que sustituir por el mayor de la izquierda, (2) si el 2-nodo tiene dos hermanos, consultar el hermano de la derecha). Realiza 1 gráfico para el borrado de cada elemento indicando las transformaciones realizadas.



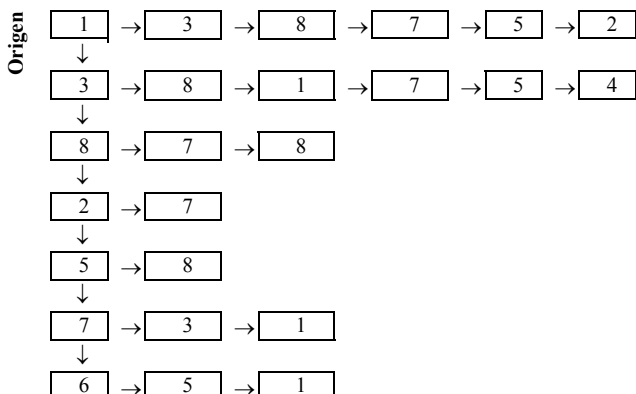
3. Dado el siguiente montículo doble borrar los elementos mínimo y máximo de forma sucesiva.

4. a) Insertar en una tabla de dispersión cerrada vacía, con estrategia de redistribución con una segunda función Hash, de tamaño B=11, la siguiente secuencia de elementos: 23, 14, 10, 15, 3, 5, 7, 8, 36, 47 y 4. Detallar la secuencia de intentos

b) Dado el grafo dirigido representado por la lista de listas que se muestra a continuación, obtener el bosque extendido en profundidad que parte del vértice 1, con su clasificación de arcos. Para recorrer las listas de adyacencia, seguir el orden de izquierda a derecha de cada lista.



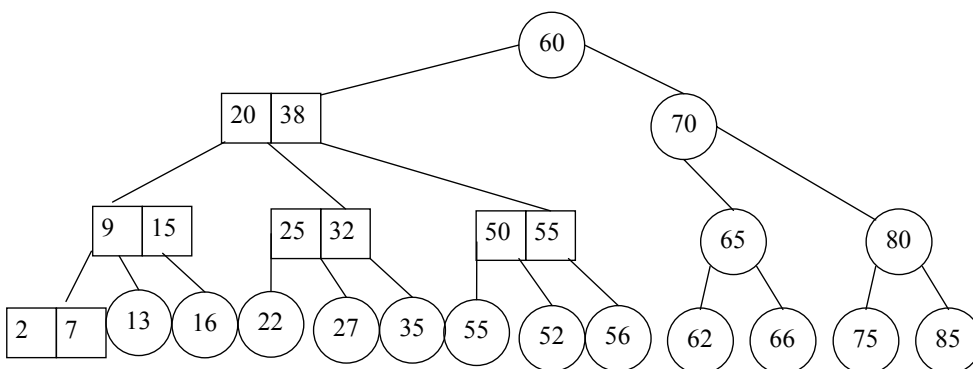
Destino



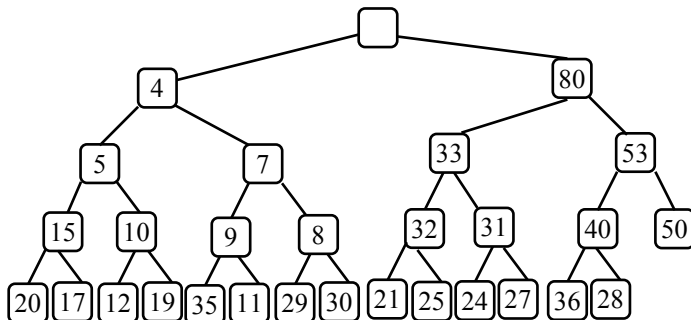
Examen TAD/PED septiembre 2005. Soluciones

1. a) obtener(lista,posicion) → item
 obtener(crear(),p) = **error_item**.....
 si p==primera (inscabeza(l₁,x)) entonces obtener(inscabeza(l₁,x),p)= **x**
 sino obtener(inscabeza(l₁,x),p)= **obtener(l₁,p)**.....
 b) resta(natural,natural) → natural
 resta (**a** , **cero**) = **a**
 resta (**suc(a)** , **suc(b)**) = **resta(a,b)**.....
2. BORRADO DEL 40: 2 COMBINACIONES Y 1 ROTACIÓN
 BORRADO DEL 30: 1 ROTACIÓN
 BORRADO DEL 12: 3 COMBINACIONES

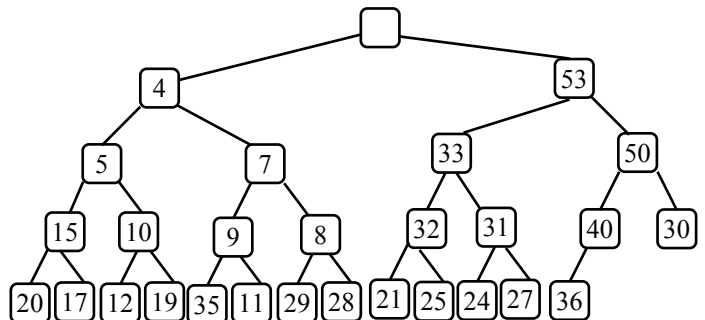
RESULTADO FINAL:



3. a) Borrado del mínimo:



b) Borrado del máximo:



4. a) $k(x) = (x \text{ MOD } (B-1)) + 1$ $h_i(x) = (H(x) + k(x) \cdot i) \text{ MOD } B$

0	1	2	3	4	5	6	7	8	9	10
47	23	4	14	15	5	36	3	8	7	10

4. b)

Recorrido en profundidad: 1, 3, 8, 7, 5, 4, 2, 6

Arcos de avance: <3, 7>, <1, 8>, <1, 7>, <1, 5>

Arcos de cruce: <5, 8>, <2, 7>, <6, 5>, <6, 1>

Arcos de retroceso: <7, 3>, <7, 1>, <3, 1>, <8, 8> **Arcos del árbol:** los restantes

