

Apellidos:

Nombre:

Convocatoria:

DNI:

## Examen PED junio 2009

### Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
  - Tiempo para efectuar el test: **17 minutos**.
  - Una pregunta mal contestada elimina una correcta.
  - Las soluciones al examen se dejarán en el campus virtual.
  - **Una vez empezado el examen no se puede salir del aula hasta finalizarlo.**
  - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

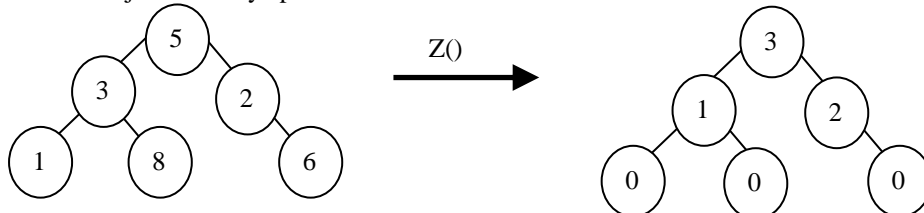
	V	F		
El operador "." se puede sobrecargar en C++	<input type="checkbox"/>	<input type="checkbox"/>	1	F
La complejidad temporal (en su caso peor) del siguiente fragmento de código es $O(n^2)$ <pre>int i, j, n, sum; for (i = 4; i &lt; n; i++) {     for (j = i-3, sum = a[i-4]; j &lt;= i; j++) sum += a[j];     cout &lt;&lt; "La suma del subarray " &lt;&lt; i-4 &lt;&lt; " es " &lt;&lt; sum &lt;&lt; endl; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	2	F
La semántica de la operación borrar del tipo lista vista en clase es la siguiente: <pre>VAR Ll: lista; x: item; p: posicion; borrar( crear( ), p ) = crear( ) si p == primera( inscabeza( Ll, x ) ) entonces     borrar( inscabeza( Ll, x ), p ) = Ll si no     borrar( inscabeza( Ll, x ), p ) = inscabeza( borrar( Ll, p ), x )</pre>	<input type="checkbox"/>	<input type="checkbox"/>	3	V
Dada la siguiente representación secuencial del árbol binario A, 14,8,19,5,0,0,21, el ítem 8 es el hijo izquierda del ítem 19	<input type="checkbox"/>	<input type="checkbox"/>	4	F
Un árbol AVL completo es también un árbol completamente equilibrado	<input type="checkbox"/>	<input type="checkbox"/>	5	F
En un árbol 2-3, la altura siempre disminuye si la raíz es de tipo 2-nodo y al efectuar el borrado de un elemento es necesario realizar una combinación con el nodo raíz	<input type="checkbox"/>	<input type="checkbox"/>	6	V
En el algoritmo del borrado de un elemento en un árbol 2-3-4 siempre que p sea 2-nodo hay que hacer una reestructuración	<input type="checkbox"/>	<input type="checkbox"/>	7	F
La estructura de un árbol 2-3-4 con $2^h-1$ elementos, donde $h$ es la altura del árbol, se corresponde con la estructura de un árbol binario de búsqueda lleno	<input type="checkbox"/>	<input type="checkbox"/>	8	V
Un árbol Rojo – Negro de altura 3 puede tener todos sus enlaces de color rojo	<input type="checkbox"/>	<input type="checkbox"/>	9	F
Un árbol Rojo – Negro con todos sus enlaces negros tiene el mismo número de claves que el árbol 2-3-4 equivalente	<input type="checkbox"/>	<input type="checkbox"/>	10	V
La siguiente especificación corresponde a la operación unión de conjuntos: <pre>VAR A, B: Conjuntos; x:item Union(crear(), A) = A Union(Insertar(A,x),B)= si (Pertenece(B,x)) entonces Union(A, B) sino Insertar(Union(A, B), x)</pre>	<input type="checkbox"/>	<input type="checkbox"/>	11	V
En el TAD Diccionario con dispersión cerrada, con función de redispersión $h_i(x)=(H(x) + k(x)*i) \text{ MOD } B$ , la función $k(x)$ podría tomar valores entre 0 y $B-1$ , siendo $B$ el tamaño de la tabla	<input type="checkbox"/>	<input type="checkbox"/>	12	F

## Examen PED junio 2009

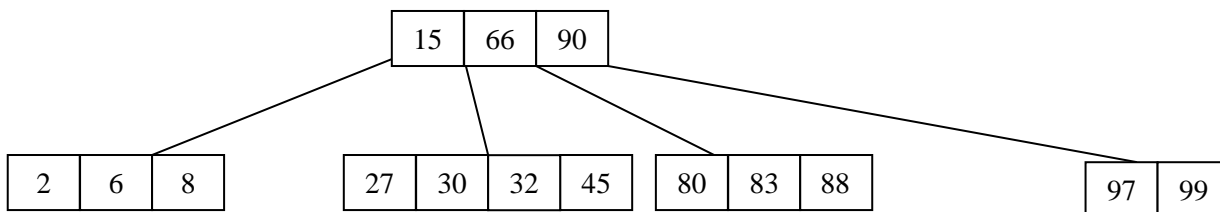
- Normas:**
- ♦ Tiempo para efectuar el ejercicio: **2 horas**
  - En la cabecera de cada hoja **Y EN ESTE ORDEN** hay que poner: **APELLIDOS, NOMBRE**.
  - Cada pregunta se escribirá en hojas diferentes.
  - Se dispone de 20 minutos para abandonar el examen sin que corra convocatoria.
  - Las soluciones 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 notas:** se publicará un anuncio en el campus virtual.

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

1. A partir de la especificación algebraica de los árboles binarios, y suponiendo que las etiquetas de los nodos son números enteros, escribe la sintaxis y semántica de la operación  $Z()$  que recibe un árbol binario en el que sustituye la etiqueta de cada nodo por la etiqueta de la raíz del subárbol izquierdo si existe (si no existiese subárbol izquierda, la etiqueta sería la misma) y en el caso de los nodos hoja la sustituye por cero:



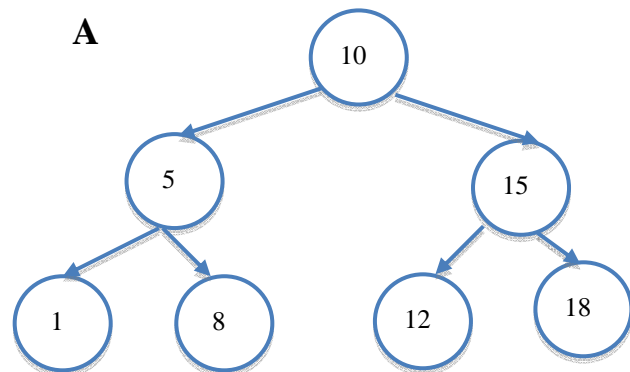
2. Suponer el árbol B de orden  $m = 5$ , mostrado a continuación. Borra el mínimo número de elementos del árbol para que el número de elementos en la raíz disminuya en uno. Muestra el resultado después de cada eliminación. Criterios: si el nodo tiene dos hermanos elegir hermano de la izquierda. Sustituir por el mayor de la izquierda.



3. Dado el siguiente árbol A:

- Si el árbol A cumple con las propiedades de árbol binario de búsqueda (ABB): explica cuáles son estas propiedades y después inserta los elementos 13, 14 y 16 utilizando el algoritmo de inserción de árboles ABB.
- Si el árbol A cumple con las propiedades de árbol AVL: explica cuáles son estas propiedades y después inserta los elementos 13, 14 y 11 utilizando el algoritmo de inserción de AVL.
- Si el árbol A cumple con las propiedades de árbol 2-3: explica cuáles son estas propiedades y después borra los elementos 12, 15 y 18 utilizando el algoritmo de borrado de árboles 23.
- Si el árbol A cumple con las propiedades de árbol 2-3-4: explica cuáles son estas propiedades y después borra los elementos 12, 15, 18 y 10 utilizando el algoritmo de borrado de los árboles 234. (Criterio: Si el nodo  $p$  tiene dos nodos adyacentes ( $r$ ), utilizaremos el nodo de la derecha como nodo adyacente).

**A**



4. Dado el grafo no dirigido representado por la lista de de adyacencia que se muestra a continuación:

- Obtener el árbol extendido en profundidad partiendo del vértice 1 y la clasificación de las aristas.
- Obtener el árbol extendido en anchura partiendo del vértice 1 y la clasificación de las aristas.

Nota: La lista de adyacencia de cada vértice se recorre de menor a mayor vértice para todos los casos del ejercicio.

1  $\rightarrow$  3  $\rightarrow$  5  $\rightarrow$  8  $\rightarrow$  10  $\rightarrow$  11  $\rightarrow$  12  
 2  $\rightarrow$  1  $\rightarrow$  4  $\rightarrow$  8  $\rightarrow$  10  $\rightarrow$  11  $\rightarrow$  12  
 3  $\rightarrow$  2  $\rightarrow$  4  $\rightarrow$  10  $\rightarrow$  11  $\rightarrow$  12  
 4  $\rightarrow$  5  $\rightarrow$  1  $\rightarrow$  8  $\rightarrow$  10  $\rightarrow$  11  $\rightarrow$  12  
 5  $\rightarrow$  2  $\rightarrow$  3  $\rightarrow$  10  $\rightarrow$  11  $\rightarrow$  12  
 6  $\rightarrow$  9  
 7  $\rightarrow$  6  $\rightarrow$  9  
 8  $\rightarrow$  5  $\rightarrow$  3  $\rightarrow$  10  $\rightarrow$  11  $\rightarrow$  12  
 10  $\rightarrow$  11  
 12  $\rightarrow$  11

## Examen PED junio 2009. Soluciones

1.

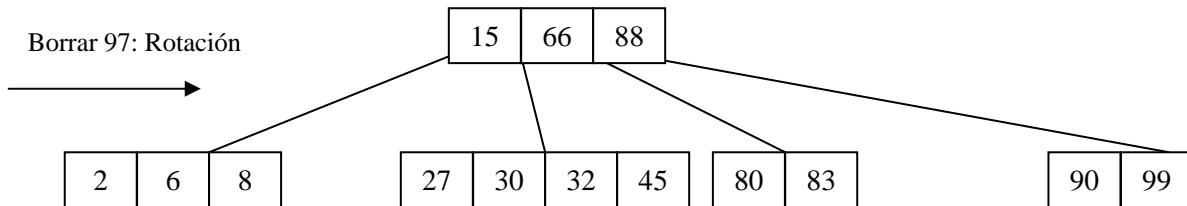
```
** Sintaxis:  
Z(arbin) → arbin  
** Semántica:  
Var i, d: arbin; x: entero;  
Z(crea_arbin()) = crea_arbin()  
Z(enraizar(crea_arbin(), x, crea_arbin())) = enraizar(crea_arbin(), 0, crea_arbin())  
si esvacio(i) entonces  
  Z(enraizar(i, x, d)) = enraizar(crea_arbin(), x, Z(d))  
sino  
  Z(enraizar(i, x, d)) = enraizar(Z(i), raiz(i), Z(d))  
fsi
```

Solución (otra forma de escribir lo mismo):

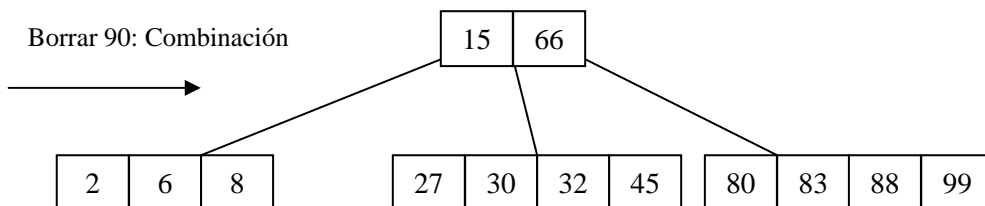
```
Var d, i1, d1: arbin; x, y: entero;  
Z(crea_arbin()) = crea_arbin()  
Z(enraizar(crea_arbin(), x, crea_arbin())) = enraizar(crea_arbin(), 0, crea_arbin())  
Z(crea_arbin(), x, d) = enraizar(crea_arbin(), x, Z(d))  
Z(enraizar(i1, y, d1), x, d) = enraizar(Z(enraizar(i1, y, d1)), y, Z(d))
```

2.

Borrar 97: Rotación



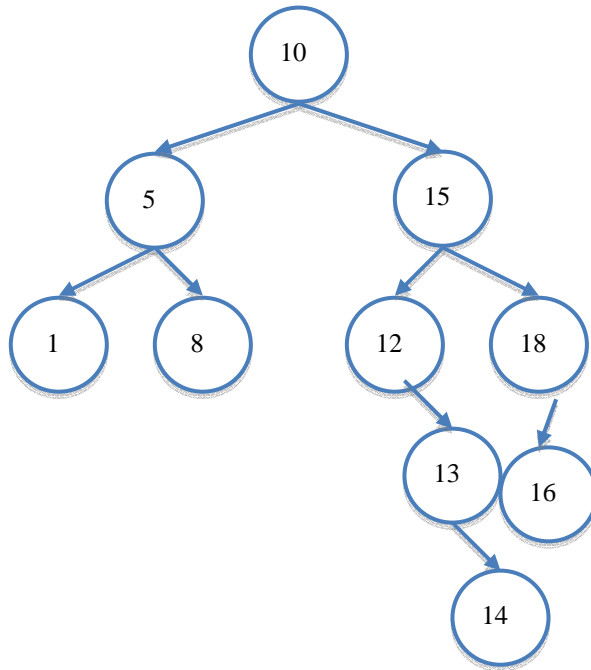
Borrar 90: Combinación



3.

a) Sí que cumple las propiedades de árbol binario de búsqueda.

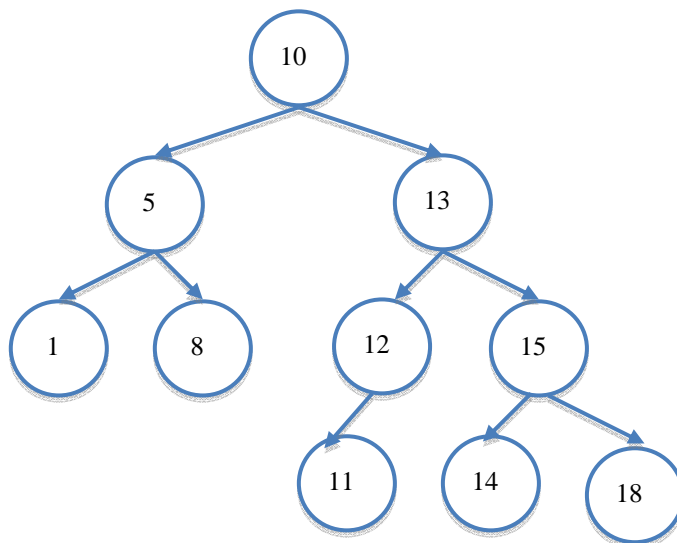
- Todos los nodos tienen un solo ítem.
- Todos los nodos cumplen las propiedades de búsqueda:
  - o todos los elementos en el subárbol izquierdo son  $\leq$  que la raíz,
  - o todos los elementos en el subárbol derecho son  $\geq$  que la raíz,
  - o los dos subárboles son binarios de búsqueda



b) Sí que cumple las propiedades de AVL.

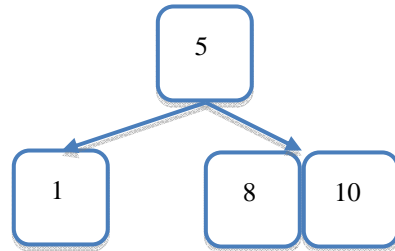
- Es un árbol binario de búsqueda.
- Es un árbol equilibrado respecto a la altura:

*“Un árbol está equilibrado respecto a la altura si y solo si para cada uno de sus nodos ocurre que las alturas de los dos subárboles difieren como mucho en 1”*



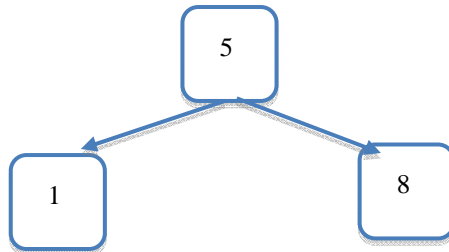
c) Sí que cumple las propiedades de árbol 23.

- Todos los nodos son de tipo 2-nodo o 3-nodo.
- Cumple las propiedades de árbol multcamino de búsqueda.
- Todas las hojas están en el mismo nivel.



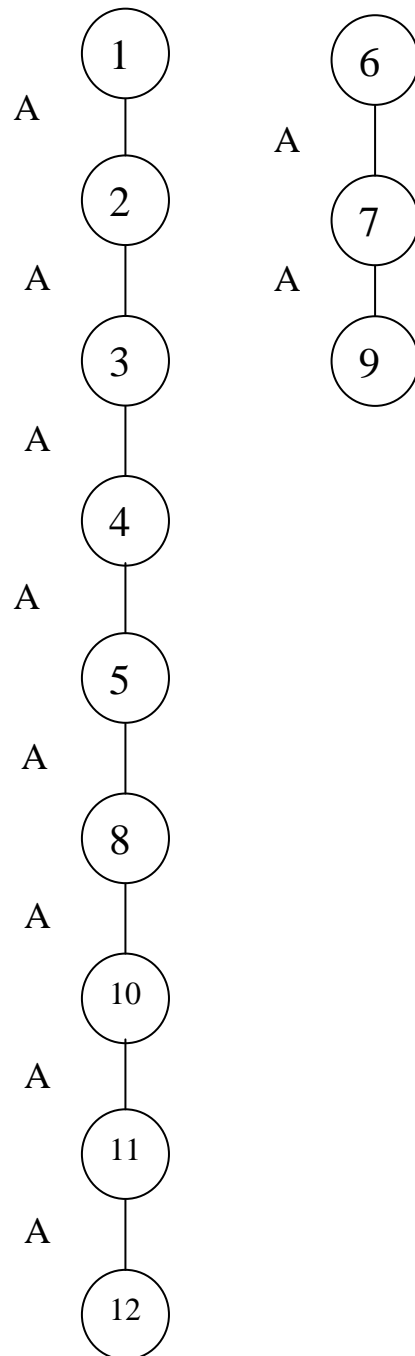
d) Sí que cumple las propiedades de árbol 234.

- Todos los nodos son de tipo 2-nodo, 3-nodo o 4-nodo.
- Cumple las propiedades de árbol multcamino de búsqueda.
- Todas las hojas están en el mismo nivel.



4.

a) Árbol extendido en profundidad. Las aristas marcadas son de árbol (A), el resto son de retroceso



b) Árbol extendido en anchura. Las aristas marcadas son de árbol (A), el resto son de retroceso

