

Apellidos, Nombre:

DNI:

Examen PED abril 2013

Modalidad 0

Normas:

- Tiempo para efectuar el test: **25 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									
En general, las operaciones modificadoras y consultoras se especifican en términos de las generadoras. En ocasiones, una operación modificadora puede especificarse en términos de otras modificadoras o consultoras. Diremos que se trata de una operación derivada.	<input type="checkbox"/>	<input type="checkbox"/>	1	V							
Sea un vector de números naturales. La operación <i>eliminar</i> que borra las posiciones pares del vector marcándolas con “0”, vista en clase, se define así: eliminar: vector -> vector Var v:vector; i: entero; x:natural; eliminar(crear()) = crear() si (i MOD 2) == 0 entonces eliminar(asignar(v,i,x)) = asignar(eliminar(v),i,x) si no eliminar(asignar(v,i,x)) = asignar(eliminar(v),i,0)	<input type="checkbox"/>	<input type="checkbox"/>	2	F							
En C++, el puntero this sólo se puede usar dentro de los métodos de la clase.	<input type="checkbox"/>	<input type="checkbox"/>	3	V							
En C++, después de invocar el destructor (~NombreClase) de un objeto, no se puede acceder a los miembros (propiedades y métodos) de dicho objeto.	<input type="checkbox"/>	<input type="checkbox"/>	4	F							
En la escala de complejidades, la mejor complejidad temporal que se puede conseguir en un algoritmo es O(n), siendo “n” la talla del problema.	<input type="checkbox"/>	<input type="checkbox"/>	5	F							
El algoritmo de búsqueda binaria estudiado en clase (búsqueda de un elemento en un vector ordenado) tiene una complejidad de $\Omega(1)$.	<input type="checkbox"/>	<input type="checkbox"/>	6	V							
La operación <i>BorrarItem</i> tiene la siguiente sintaxis y semántica: BorrarItem: LISTA, ITEM -> LISTA BorrarItem(Crear, i) = Crear BorrarItem(IC(L1,j), i) = si (i == j) entonces L1 sino IC (BorrarItem (L1, i), j) Esta operación borra la primera ocurrencia del item que se encuentra en la lista.	<input type="checkbox"/>	<input type="checkbox"/>	7	V							
La operación <i>base</i> , vista en clase, que actúa sobre una pila y devuelve la base de la pila (el primer elemento que se ha apilado) es la siguiente: base(pila) -> item Var p: pila; x: item; base(crear()) = error() base(apilar(crear(),x)) = x base(apilar(p,x)) = base(desapilar(p))	<input type="checkbox"/>	<input type="checkbox"/>	8	F							
Dado un único recorrido de un árbol binario, es posible reconstruir dicho árbol.	<input type="checkbox"/>	<input type="checkbox"/>	9	F							
El grado de un nodo es el número máximo de items asociados a dicho nodo.	<input type="checkbox"/>	<input type="checkbox"/>	10	F							
El coste temporal (en su peor caso) de insertar una etiqueta en un árbol binario de búsqueda es lineal respecto al número de nodos del árbol.	<input type="checkbox"/>	<input type="checkbox"/>	11	V							
Dada la siguiente representación secuencial del árbol binario A, <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>14</td><td>8</td><td>19</td><td>5</td><td></td><td></td><td></td></tr></table> el elemento 5 es el hijo izquierda del elemento 8.	14	8	19	5				<input type="checkbox"/>	<input type="checkbox"/>	12	V
14	8	19	5								
Cuando se realiza un borrado en un árbol AVL, en el camino de vuelta atrás para actualizar los factores de equilibrio, como mucho sólo se va a efectuar una rotación.	<input type="checkbox"/>	<input type="checkbox"/>	13	F							
El número mínimo de nodos que tiene un árbol AVL de altura 4 es 7.	<input type="checkbox"/>	<input type="checkbox"/>	14	V							

Apellidos, Nombre:

DNI:

Examen PED marzo 2015

Modalidad 0

Normas:

- Tiempo para efectuar el test: **20 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		
En la especificación algebraica, las operaciones constructoras se clasifican en generadoras y modificadoras.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
Las ecuaciones (vistas en clase) que permiten realizar la suma de números naturales son las siguientes: VAR x, y: natural; suma(x, cero) = x suma(cero, x) = x suma(x, suc(y)) = suma(x, y)	<input type="checkbox"/>	<input type="checkbox"/>	2	F
Dentro de la especificación algebraica de los números naturales, definimos la sintaxis de la función F como: F: natural → BOOL, y su semántica como: F(cero)=TRUE, F(suc(cero))=FALSE, F(suc(suc(x)))=F(x). Para el número natural x=35, la función F devolvería FALSE. Nota: se asume que x=35 es la forma simplificada de indicar x=suc(suc(suc(.....suc(cero).....))).	<input type="checkbox"/>	<input type="checkbox"/>	3	V
Todo árbol binario de altura 9 y 511 nodos es un árbol binario lleno y además es árbol binario de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	4	F
Sea el método Primera perteneciente a la clase TLista que devuelve la primera posición de la lista que lo invoca: <pre>TPosicion TLista::Primera() { TPosicion p; p.pos = primero; return p; }</pre> <pre>class TLista { public: ... private: TNode *primero; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	5	F
En el método Primera, se invoca de forma implícita a los constructores de TPosicion y TLista.				
En C++, si la variable p es un puntero a un objeto, entonces la expresión p.f() es sintácticamente correcta.	<input type="checkbox"/>	<input type="checkbox"/>	6	F
La complejidad temporal del siguiente fragmento de código es O(n) int i, j, n, sum; for (i = 4; i < n; i++) { for (j = i-3, sum = a[i-4]; j <= i; j++) sum += a[j]; cout << "La suma del subarray " << i-4 << " es " << sum << endl; }	<input type="checkbox"/>	<input type="checkbox"/>	7	V
La mejor complejidad temporal que se puede conseguir en un algoritmo es O(n), siendo "n" la talla del problema.	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Es posible reconstruir un único árbol binario de búsqueda a partir de su recorrido en postorden	<input type="checkbox"/>	<input type="checkbox"/>	9	V
El máximo número de nodos en un nivel i-1 de un árbol binario es 2^{i-2} , $i \geq 2$	<input type="checkbox"/>	<input type="checkbox"/>	10	V
Un camino en un árbol es una secuencia a_1, \dots, a_s de árboles tal que para todo $i \in \{1, \dots, s-1\}$, a_{i+1} es subárbol de a_i .	<input type="checkbox"/>	<input type="checkbox"/>	11	V
El grado de un árbol es el máximo nivel que pueden tener sus subárboles.	<input type="checkbox"/>	<input type="checkbox"/>	12	F
La operación desencolar vista en clase es la siguiente: VAR c: cola, x: item; desencolar(crear()) = crear() si esvacía(c) entonces desencolar(encolar(c, x)) = crear() si no desencolar(encolar(c, x)) = encolar(desencolar(c), x)	<input type="checkbox"/>	<input type="checkbox"/>	13	V
El ítem medio (según la relación de orden en la búsqueda) almacenado en un árbol binario de búsqueda siempre se encuentra en la raíz.	<input type="checkbox"/>	<input type="checkbox"/>	14	F

Apellidos, Nombre:

DNI:

Examen PED abril 2016

Modalidad 0

Normas:

- Tiempo para efectuar el test: **20 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		
En la especificación algebraica, una operación es una función que toma como parámetros (entrada) uno o más valores de diversos tipos, y produce como resultado un solo valor de otro tipo.	<input type="checkbox"/>	<input type="checkbox"/>	1	F
Las ecuaciones (vistas en clase) que permiten realizar la multiplicación de números naturales son las siguientes: VAR x, y: natural; mult(cero, x) = cero mult(x, cero) = cero mult(suc(y), x) = suma(mult(y, x), x)	<input type="checkbox"/>	<input type="checkbox"/>	2	V
En la especificación algebraica, para el tratamiento de errores se añade una constate a la signatura que modeliza un valor de error, por ejemplo $\text{error}_{\text{nat}} \rightarrow \text{natural}$.	<input type="checkbox"/>	<input type="checkbox"/>	3	V
En C++, si se declara un objeto <i>a</i> (p. ej. <i>TPorro a;</i>) cuando la variable <i>a</i> se sale de ámbito entonces se invoca automáticamente al destructor de ese objeto.	<input type="checkbox"/>	<input type="checkbox"/>	4	V
Las ecuaciones (vistas en clase) para la operación <i>recu</i> de un vector son las siguientes: recu(crear(), i) = error() recu(asig(v, i, x), j) si (i == j) entonces j si no recu(v, j) fsi	<input type="checkbox"/>	<input type="checkbox"/>	5	F
La complejidad temporal de la operación <i>desapilar</i> (vista en clase) utilizando vectores (con un índice que indica la cima de la pila) o utilizando listas enlazadas es la misma.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
La complejidad temporal del siguiente fragmento de código es $O(n^2)$ int i, j, n, sum; for (i = 4; i < n; i++) { for (j = i-3, sum = a[i-4]; j <= i; j++) sum += a[j]; cout << "La suma del subarray " << i-4 << " es " << sum << endl; }	<input type="checkbox"/>	<input type="checkbox"/>	7	F
En las colas circulares enlazadas vistas en clase, las operaciones <i>encolar</i> y <i>desencolar</i> tienen complejidad temporal $\Theta(1)$.	<input type="checkbox"/>	<input type="checkbox"/>	8	V
Las ecuaciones (vistas en clase) para la operación <i>desencolar</i> son las siguientes: desencolar(crear()) = crear() si esvacía(c) entonces desencolar(encolar(c, x)) = crear() si no desencolar(encolar(c, x)) = encolar(desencolar(c), x)	<input type="checkbox"/>	<input type="checkbox"/>	9	V
Es posible reconstruir un único árbol binario de búsqueda a partir de un recorrido en preorden.	<input type="checkbox"/>	<input type="checkbox"/>	10	V
Un camino en un árbol es una secuencia a_1, \dots, a_s de árboles tal que para todo $i \in \{1, \dots, s-1\}$, a_i es subárbol de a_{i+1} .	<input type="checkbox"/>	<input type="checkbox"/>	11	F
A los árboles generales también se les llama árboles multicamino de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	12	F
La semántica de la operación <i>quita_hojas</i> que actúa sobre un árbol binario y devuelve el árbol binario original sin sus hojas es la siguiente: VAR i, d: arbin; x: item; quita_hojas(crea_arbin()) = crea_arbin() quita_hojas(enraizar(crea_arbin(), x, crea_arbin())) = enraizar(crea_arbin(), x, crea_arbin()) quita_hojas(enraizar(i, x, d)) = enraizar(quita_hojas(i), x, quita_hojas(d))	<input type="checkbox"/>	<input type="checkbox"/>	13	F
Profundidad de un subárbol es la longitud del único camino desde la raíz a dicho subárbol.	<input type="checkbox"/>	<input type="checkbox"/>	14	V

Apellidos, Nombre:

DNI:

Examen PED abril 2013

Modalidad 2

Normas:

- * Tiempo para efectuar el test: **25 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**.

En la hoja de contestaciones el verdadero se corresponderá con la A, y el falso con la B.											
	V	F									
La operación <i>base</i> , vista en clase, que actúa sobre una pila y devuelve la base de la pila (el primer elemento que se ha apilado) es la siguiente: base(pila) -> item Var p: pila; x: item; base(crear()) = error() base(apilar(crear(),x)) = x base(apilar(p,x)) = base(desapilar(p))	<input type="checkbox"/>	<input type="checkbox"/>	1	F							
Sea un vector de números naturales. La operación <i>eliminar</i> que borra las posiciones pares del vector marcándolas con “0”, vista en clase, se define así: eliminar: vector -> vector Var v:vector; i: entero; x:natural; eliminar(crear()) = crear() si (i MOD 2) == 0 entonces eliminar(asignar(v,i,x)) = asignar(eliminar(v),i,x) si no eliminar(asignar(v,i,x)) = asignar(eliminar(v),i,0)	<input type="checkbox"/>	<input type="checkbox"/>	2	F							
El algoritmo de búsqueda binaria estudiado en clase (búsqueda de un elemento en un vector ordenado) tiene una complejidad de $\Omega(1)$.	<input type="checkbox"/>	<input type="checkbox"/>	3	V							
En C++, después de invocar el destructor (~NombreClase) de un objeto, no se puede acceder a los miembros (propiedades y métodos) de dicho objeto.	<input type="checkbox"/>	<input type="checkbox"/>	4	F							
En la escala de complejidades, la mejor complejidad temporal que se puede conseguir en un algoritmo es O(n), siendo “n” la talla del problema.	<input type="checkbox"/>	<input type="checkbox"/>	5	F							
En C++, el puntero this sólo se puede usar dentro de los métodos de la clase.	<input type="checkbox"/>	<input type="checkbox"/>	6	V							
La operación <i>BorrarItem</i> tiene la siguiente sintaxis y semántica: BorrarItem: LISTA, ITEM -> LISTA BorrarItem(Crear, i) = Crear BorrarItem(IC(L1,j), i) = si (i == j) entonces L1 sino IC (BorrarItem (L1, i), j) Esta operación borra la primera ocurrencia del item que se encuentra en la lista.	<input type="checkbox"/>	<input type="checkbox"/>	7	V							
En general, las operaciones modificadoras y consultoras se especifican en términos de las generadoras. En ocasiones, una operación modificadora puede especificarse en términos de otras modificadoras o consultoras. Diremos que se trata de una operación derivada.	<input type="checkbox"/>	<input type="checkbox"/>	8	V							
Dado un único recorrido de un árbol binario, es posible reconstruir dicho árbol.	<input type="checkbox"/>	<input type="checkbox"/>	9	F							
Cuando se realiza un borrado en un árbol AVL, en el camino de vuelta atrás para actualizar los factores de equilibrio, como mucho sólo se va a efectuar una rotación.	<input type="checkbox"/>	<input type="checkbox"/>	10	F							
El coste temporal (en su peor caso) de insertar una etiqueta en un árbol binario de búsqueda es lineal respecto al número de nodos del árbol.	<input type="checkbox"/>	<input type="checkbox"/>	11	V							
Dada la siguiente representación secuencial del árbol binario A, <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>14</td><td>8</td><td>19</td><td>5</td><td></td><td></td><td></td></tr></table> el elemento 5 es el hijo izquierda del elemento 8.	14	8	19	5				<input type="checkbox"/>	<input type="checkbox"/>	12	V
14	8	19	5								
El grado de un nodo es el número máximo de ítems asociados a dicho nodo.	<input type="checkbox"/>	<input type="checkbox"/>	13	F							
El número mínimo de nodos que tiene un árbol AVL de altura 4 es 7.	<input type="checkbox"/>	<input type="checkbox"/>	14	V							

Apellidos, Nombre:

DNI:

Examen PED abril 2014

Modalidad 0

Normas:

- Tiempo para efectuar el test: **20 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		
La operación <i>palindromo</i> (sobre un vector) vista en clase es la siguiente: Var v: vector; i,x: natural; palindromo(crear_vector()) = VERDADERO palindromo(asig(v,i,x)) = si $i \leq 50$ entonces si $\text{recu}(v, 100-i+1) == x$ entonces palindromo(asig(v,i,x)) sino FALSO sino VERDADERO	<input type="checkbox"/>	<input type="checkbox"/>	1	F
Una operación del TAD X que tenga la sintaxis $\text{Crear()} \rightarrow X$ es una operación constructora generadora.	<input type="checkbox"/>	<input type="checkbox"/>	2	V
En C++, al hacer layering los métodos de la clase derivada pueden acceder a la parte pública y privada de la clase base.	<input type="checkbox"/>	<input type="checkbox"/>	3	F
En C++, si un objeto se sale de ámbito entonces se invoca automáticamente al destructor de ese objeto.	<input type="checkbox"/>	<input type="checkbox"/>	4	V
El algoritmo de intercambio directo o burbuja estudiado en clase (ordenación de los elementos de un vector) tiene una complejidad promedio de $\Theta(n^2)$, siendo n el número de elementos del vector.	<input type="checkbox"/>	<input type="checkbox"/>	5	V
La complejidad espacial es la cantidad de recursos espaciales que un algoritmo consume o necesita para su ejecución.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
La complejidad temporal en el peor caso para la inserción de un elemento en una lista ordenada y en otra no ordenada, que no permiten elementos repetidos, siempre es lineal con el número de elementos en ambos casos.	<input type="checkbox"/>	<input type="checkbox"/>	7	V
El tipo de datos vector (visto en clase) se define como un conjunto en el que sus componentes ocupan posiciones consecutivas de memoria.	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Sea el TIPO arbin definido en clase. La semántica de la operación nodos es la siguiente: Var i,d:arbin; x:item; nodos(crear_arbin())=0 nodos(enraizar(i,x,d))=nodos(i)+nodos(d)	<input type="checkbox"/>	<input type="checkbox"/>	9	F
Es posible reconstruir un único árbol binario de altura 6 a partir de un recorrido en postorden con 63 etiquetas.	<input type="checkbox"/>	<input type="checkbox"/>	10	V
El ítem medio (según la relación de orden) almacenado en un árbol binario de búsqueda lleno siempre se encuentra en la raíz.	<input type="checkbox"/>	<input type="checkbox"/>	11	V
A los árboles generales también se les llama árboles multicamino de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	12	F
Las rotaciones que hay que realizar en los árboles AVL para mantenerlos balanceados tienen un coste temporal (en su peor caso) lineal con respecto al número de ítems del árbol.	<input type="checkbox"/>	<input type="checkbox"/>	13	F
El grado de los árboles AVL puede ser +1, 0 ó -1.	<input type="checkbox"/>	<input type="checkbox"/>	14	F

Apellidos:

Nombre:

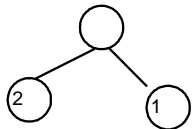
Convocatoria:

DNI:

Examen TAD/PED diciembre 2003

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
 - **El test vale un 40% de la nota de teoría.**
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F	
La complejidad logarítmica aparece en algoritmos que descartan muchos valores (generalmente la mitad) en un único paso.	<input type="checkbox"/>	<input type="checkbox"/>	V 1.
En C++, la parte privada de una clase sólo es accesible por los métodos de la propia clase.	<input type="checkbox"/>	<input type="checkbox"/>	F 2.
El tipo posición en una lista con acceso por posición se puede instanciar a diferentes tipos de objetos.	<input type="checkbox"/>	<input type="checkbox"/>	V 3.
El recorrido en postorden es el inverso especular del recorrido en preorden para un árbol binario dado.	<input type="checkbox"/>	<input type="checkbox"/>	V 4.
En un árbol AVL las inserciones siempre se realizan en las hojas.	<input type="checkbox"/>	<input type="checkbox"/>	V 5.
Dado un árbol 2-3 de altura h con n items con todos sus nodos del tipo 3-Nodo: la complejidad de la operación de búsqueda de un ítem es $O(\log_3 h)$.	<input type="checkbox"/>	<input type="checkbox"/>	F 6.
En un árbol 2-3-4 las reestructuraciones se realizan desde la raíz hacia las hojas.	<input type="checkbox"/>	<input type="checkbox"/>	V 7.
En un árbol rojo-negro, el número de enlaces negros ha de ser mayor que el de enlaces rojos.	<input type="checkbox"/>	<input type="checkbox"/>	F 8.
La altura del árbol B m-camino de búsqueda es " $\log_m n$ ", con " n =número total de claves".	<input type="checkbox"/>	<input type="checkbox"/>	F 9.
La función de redispersión en una tabla hash abierta, para que se recorran todas las posiciones del vector, tiene que cumplir que el valor de B sea primo.	<input type="checkbox"/>	<input type="checkbox"/>	F 10.
El siguiente árbol es un montículo doble: 	<input type="checkbox"/>	<input type="checkbox"/>	F 11.
Todo árbol Leftist cumple las condiciones para ser un árbol binario de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	F 12.
Al representar un grafo dirigido de N vértices y K aristas con una lista de adyacencia, la operación de hallar la adyacencia de entrada de un vértice, tiene una complejidad de $O(N^2)$.	<input type="checkbox"/>	<input type="checkbox"/>	V 13.
En C++, el constructor de copia sustituye al operador asignación.	<input type="checkbox"/>	<input type="checkbox"/>	F 14.

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen TAD/PED diciembre 2004

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
 - **El test vale un 40% de la nota de teoría.**
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F	
En un Tipo Abstracto de Datos la manipulación de los objetos o valores de un tipo, sólo depende del comportamiento descrito en su especificación y es independiente de su implementación.	<input type="checkbox"/>	<input type="checkbox"/>	V 1.
La complejidad temporal (en su peor caso) de la operación de insertar un elemento en una cola circular enlazada que no admite elementos repetidos es $O(n)$.	<input type="checkbox"/>	<input type="checkbox"/>	V 2.
La complejidad temporal de la operación <i>apilar</i> en una pila es independiente de su implementación.	<input type="checkbox"/>	<input type="checkbox"/>	F 3.
Sea el TIPO <i>arbin</i> definido en clase. La semántica de la operación <i>nodos</i> es la siguiente: <i>var i,d:arbin; x:item;</i> <i>nodos(crear_arbin())=0</i> <i>nodos(enraizar(i,x,d))=nodos(i)+nodos(d)</i>	<input type="checkbox"/>	<input type="checkbox"/>	F 4.
En un árbol AVL, el factor de equilibrio de un nodo se define como el valor absoluto de la altura del subárbol de la derecha menos la altura del subárbol de la izquierda.	<input type="checkbox"/>	<input type="checkbox"/>	F 5.
El mínimo número de elementos que se puede almacenar en un árbol 2-3 de altura <i>h</i> coincide con el número de elementos que hay en un árbol binario lleno de altura <i>h</i> .	<input type="checkbox"/>	<input type="checkbox"/>	V 6.
Al insertar un único ítem en un árbol 2-3-4 puede ocurrir que haya que realizar dos operaciones: DIVIDERAIZ (p) y DIVIDEHIJODE2 (p, q).	<input type="checkbox"/>	<input type="checkbox"/>	V 7.
La inserción de una etiqueta en un árbol rojo-negro se efectuará creando un hijo de color rojo.	<input type="checkbox"/>	<input type="checkbox"/>	V 8.
Un árbol binario de búsqueda con altura 7 y 127 nodos es un árbol B con $m=3$	<input type="checkbox"/>	<input type="checkbox"/>	V 9.
En la operación de búsqueda de un elemento en un conjunto ordenado representado mediante un vector de elementos ordenado (representación no enlazada de una lista), se puede utilizar el algoritmo de búsqueda binaria.	<input type="checkbox"/>	<input type="checkbox"/>	V 10.
El factor de carga de la dispersión abierta siempre está entre 0 y 1.	<input type="checkbox"/>	<input type="checkbox"/>	F 11.
En un montículo doble, un elemento “i” del montículo mínimo tiene como máximo un elemento simétrico “j” del montículo máximo.	<input type="checkbox"/>	<input type="checkbox"/>	V 12.
Un árbol binario de búsqueda cumple las propiedades de un árbol Leftist.	<input type="checkbox"/>	<input type="checkbox"/>	F 13.
La siguiente secuencia de nodos de un grafo es un ciclo: 1,2,3,1.	<input type="checkbox"/>	<input type="checkbox"/>	V 14.

Examen PED diciembre 2007

Modalidad 0

- Normas:**
- La entrega del test **no** corre convocatoria.
 - Tiempo para efectuar el test: **15 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		
Las ecuaciones (vistas en clase) que permiten realizar la suma de números naturales son las siguientes: VAR x, y: natural; suma(x, cero) = x suma(cero, x) = x suma(x, suc(y)) = suma(suc(x), y)	<input type="checkbox"/>	<input type="checkbox"/>	1	F
En C++, la siguiente declaración es INCORRECTA : const int& a = 1;	<input type="checkbox"/>	<input type="checkbox"/>	2	F
Dadas las clases TDir y TVectorDir con todos sus métodos implementados: constructor, constructor sobrecargado, sobrecarga del corchete, sobrecarga del operador salida (se muestra el contenido de cada posición del vector dejando un espacio), etc. Nota: El constructor por defecto crea un vector vacío. El constructor a partir de una dimensión, pone todos los elementos a 0.	<input type="checkbox"/>	<input type="checkbox"/>	3	F
<pre> class TDir class TVectorDir main() { public: private: int e1; int e2; }; class TVectorDir { public: private: TDir *vector; int longitud; }; main() { TDir a(1,1); TVectorDir v; cout<<"v_antes="<<v<<endl; v[1]=a; cout<<"v_despues="<<v<<endl; } </pre> El resultado obtenido tras la ejecución del main() sería: v_antes= 0 0 v_despues= 1 1	<input type="checkbox"/>	<input type="checkbox"/>	4	V
La complejidad espacial es la cantidad de recursos espaciales que un algoritmo consume o necesita para su ejecución.	<input type="checkbox"/>	<input type="checkbox"/>	5	F
La complejidad temporal de un algoritmo depende de la complejidad espacial del mismo.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
La semántica de la operación <i>desencolar</i> vista en clase es la siguiente: VAR c: cola, x: item; si esvacía(c) entonces desencolar(encolar(c, x)) = crear_cola () si no desencolar(encolar(c, x)) = encolar(desencolar (c), x)	<input type="checkbox"/>	<input type="checkbox"/>	7	V
Un árbol con un único nodo es un árbol lleno.	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Un árbol con un único nodo tiene un único camino cuya longitud es 1.	<input type="checkbox"/>	<input type="checkbox"/>	9	V
Dados los recorridos de preorden, postorden y niveles de un árbol binario de altura 7 y 64 hojas es posible reconstruir un único árbol binario.	<input type="checkbox"/>	<input type="checkbox"/>	10	V
En la representación de conjuntos mediante listas, la complejidad espacial es proporcional al tamaño del conjunto representado.	<input type="checkbox"/>	<input type="checkbox"/>	11	F
En un grafo dirigido pueden existir infinitas aristas para un número "n" de vértices.	<input type="checkbox"/>	<input type="checkbox"/>		

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED diciembre 2008

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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		
Los enriquecimientos forman parte de la definición de un TAD.	<input type="checkbox"/>	<input type="checkbox"/>	1	F
Sea el método <i>Primera</i> perteneciente a la clase <i>TLista</i> que devuelve la primera posición de la lista que lo invoca: <pre>TPosicion TLista::Primera() { TPosicion p; p.pos = lis; return p; }</pre> <pre>class TLista { public: ... private: TNodeo *lis; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	2	F
En el método <i>Primera</i> , se invoca a la sobrecarga del operador asignación entre objetos del tipo <i>TPosicion</i> .				
En C++, las funciones y clases AMIGAS es obligatorio declararlas siempre antes de la sección PUBLIC de la clase	<input type="checkbox"/>	<input type="checkbox"/>	3	F
En la escala de complejidades se cumple que $O(\log n) \subset O(\log \log n)$.	<input type="checkbox"/>	<input type="checkbox"/>	4	F
Sea el tipo <i>cola</i> definido en clase. La semántica de la operación <i>cabeza</i> es la siguiente: <pre>Var c:cola; x:item; cabeza(crear_cola())=error_item() si esvacía(c) entonces cabeza(encolar(c,x))=x sino cabeza(encolar(c,x))=encolar(cabeza(c),x)</pre>	<input type="checkbox"/>	<input type="checkbox"/>	5	F
El nivel de un nodo en un árbol coincide con la longitud del camino desde la raíz a dicho nodo	<input type="checkbox"/>	<input type="checkbox"/>	6	F
A los árboles generales también se les llama árboles multicamino de búsqueda	<input type="checkbox"/>	<input type="checkbox"/>	7	F
Un árbol completo siempre está balanceado respecto a la altura	<input type="checkbox"/>	<input type="checkbox"/>	8	V
El número mínimo de elementos que se pueden almacenar en un árbol 2-3 de altura h es $2^h - 1$	<input type="checkbox"/>	<input type="checkbox"/>	9	V
Para que decrezca la altura de un árbol 2-3-4 en una operación de borrado, el nodo raíz y sus hijos tienen que ser 2-nodo	<input type="checkbox"/>	<input type="checkbox"/>	10	V
Un árbol rojo-negro es un árbol binario balanceado respecto a la altura	<input type="checkbox"/>	<input type="checkbox"/>	11	F
El árbol 2-3 es un árbol B m-camino de búsqueda con $m=2$	<input type="checkbox"/>	<input type="checkbox"/>	12	F
Sea una tabla de dispersión cerrada con estrategia de redispersión $h_i(x) = (H(x) + C \cdot i) \text{ MOD } B$, con $B=1000$ y $C=74$. Para cualquier clave "x" se recorrerán todas las posiciones de la tabla buscando una posición libre.	<input type="checkbox"/>	<input type="checkbox"/>	13	F
Para todo nodo de un árbol Leftist, se cumple que el número de nodos de su hijo izquierdo es menor que el de su hijo derecho.	<input type="checkbox"/>	<input type="checkbox"/>	14	F
En un multigrafo pueden existir infinitas aristas para un número "n" de vértices.	<input type="checkbox"/>	<input type="checkbox"/>	15	V

Apellidos:

Nombre:

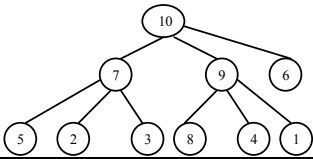
Convocatoria:

DNI:

Examen PED diciembre 2009

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **22 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		
Para el siguiente fragmento de código C++ de un posible método perteneciente a la conocida clase TCoordenada, la línea "delete b;" liberaría correctamente la memoria dinámica de b. <pre>void Funcion(void) { TCoordenada *a = new TCoordenada; TCoordenada *b = new TCoordenada[5]; (.....) delete b; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	1	F
El resultado del cálculo de la complejidad temporal en el mejor caso de un algoritmo X, da como resultado $n + n \cdot \log(n)$. Por lo tanto, diremos que la complejidad del algoritmo X cuando $n \rightarrow \infty$ pertenece a $\Omega(n)$.	<input type="checkbox"/>	<input type="checkbox"/>	2	F
Las pilas también se conocen como listas LIFO.	<input type="checkbox"/>	<input type="checkbox"/>	3	V
Dado un único recorrido de un árbol binario lleno, es posible reconstruir dicho árbol.	<input type="checkbox"/>	<input type="checkbox"/>	4	V
A los árboles generales también se les llama árboles multicamino de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	5	F
Cuando se realiza una inserción en un AVL, en el camino de vuelta atrás para actualizar los factores de equilibrio, como mucho solo se va a efectuar una rotación.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
La altura de un árbol 2-3 únicamente crece cuando se inserta un elemento y todos los nodos del árbol son 3-nodo.	<input type="checkbox"/>	<input type="checkbox"/>	7	F
Con las operaciones de inserción y borrado es posible conseguir un árbol 2-3-4 de altura 4 con todos sus nodos de tipo 2-nodo.	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Las operaciones de transformación cuando se inserta un elemento en un árbol 2-3-4, en el caso de un árbol rojo-negro, se reducen a cambios de colores o rotaciones.	<input type="checkbox"/>	<input type="checkbox"/>	9	V
El árbol 2-3 es un árbol B m-camino de búsqueda con $m=2$.	<input type="checkbox"/>	<input type="checkbox"/>	10	F
La dispersión abierta elimina el problema del clustering secundario.	<input type="checkbox"/>	<input type="checkbox"/>	11	V
Sea una tabla de dispersión cerrada con estrategia de redispersión $h_i(x) = (H(x) + C \cdot i) \text{ MOD } B$, con $B=1000$ y $C=74$. Para cualquier clave "x" se recorrerán todas las posiciones de la tabla buscando una posición libre cuando se inserta el elemento.	<input type="checkbox"/>	<input type="checkbox"/>	12	F
El siguiente árbol es un montículo máximo: 	<input type="checkbox"/>	<input type="checkbox"/>	13	F
Para todo nodo de un árbol Leftist, se cumple que el número de nodos de su hijo izquierdo es mayor o igual que el de su hijo derecho.	<input type="checkbox"/>	<input type="checkbox"/>	14	F
Un grafo no dirigido de n vértices es un árbol si está libre de ciclos y tiene $n-1$ aristas.	<input type="checkbox"/>	<input type="checkbox"/>	15	V

DNI:

Normas:

- La entrega del test no corre convocatoria.
- Tiempo para efectuar el test: **20 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		
Sea el método Primera perteneciente a la clase TLista que devuelve la primera posición de la lista que lo invoca: $TPosicion\ TLista::Primera() \quad \text{class } TLista \{$ $\{ \quad TPosicion\ p; \quad \quad \quad \text{public: } \dots$ $\quad \quad \quad \text{p.pos} = lis; \quad \quad \quad \text{private:}$ $\quad \quad \quad \text{return } p; \} \quad \quad \quad \text{TNode } *lis; \}$	<input type="checkbox"/>	<input type="checkbox"/>	1	F
En la línea resaltada, se invoca a la sobrecarga del operador asignación entre objetos del tipo TPosicion.				
En la escala de complejidades se cumple que $O(n \log n) \subset O(n^2)$.	<input type="checkbox"/>	<input type="checkbox"/>	2	V
La operación BorrarResultem tiene la siguiente sintaxis y semántica: BorrarResultem: LISTA, ITEM -> LISTA BorrarResultem(Crear, i) = Crear BorrarResultem(IC(L1,j), i) = si (i == j) entonces L1 sino IC (BorrarResultem (L1, i), j)	<input type="checkbox"/>	<input type="checkbox"/>	3	V
Esta operación borra la primera ocurrencia del ítem que se encuentra en la lista				
El nivel de un nodo en un árbol coincide con la longitud del camino desde la raíz a dicho nodo	<input type="checkbox"/>	<input type="checkbox"/>	4	F
A los árboles generales también se les llama árboles multicamino de búsqueda	<input type="checkbox"/>	<input type="checkbox"/>	5	F
Cuando se realiza una inserción en un AVL, en el camino de vuelta atrás para actualizar los factores de equilibrio, sólo se va a efectuar una rotación como mucho	<input type="checkbox"/>	<input type="checkbox"/>	6	V
Dado un árbol 2-3 con n ítems con todos sus nodos del tipo 2-Nodo: la complejidad de la operación de búsqueda de un ítem es $O(\log_2 n)$	<input type="checkbox"/>	<input type="checkbox"/>	7	V
Un árbol Rojo-Negro es una representación sobre árbol binario de un árbol 2-3	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Un árbol rojo-negro es un árbol binario balanceado respecto a la altura	<input type="checkbox"/>	<input type="checkbox"/>	9	F
La raíz del árbol B m-camino de búsqueda siempre tiene al menos $m/2$ claves o etiquetas	<input type="checkbox"/>	<input type="checkbox"/>	10	F
La especificación algebraica de la siguiente operación eliminaría todas las claves repetidas de un determinado ítem (C: ConjuntoConClavesRepetidas; x, y: Ítem): $Eliminar(Crear, x) \Leftrightarrow Crear$ $Eliminar(Insertar(C, x), y) \Leftrightarrow$ $si\ (x == y)\ \text{entonces } C\ \text{sino } Insertar(Eliminar(C, y), x)$	<input type="checkbox"/>	<input type="checkbox"/>	11	F
En una tabla de dispersión cerrada con la siguiente función de redispersión para la clave 14: $h_i(14) = (28 + 7*i) \text{ MOD } 2000$, se recorrerán todas las posiciones de la tabla buscando una posición libre.	<input type="checkbox"/>	<input type="checkbox"/>	12	V
El TAD Cola de Prioridad representado por un montículo, tendrá las siguientes complejidades: $O(1)$ para el borrado, y $O(\log n)$ para la inserción, siendo n el número de elementos.	<input type="checkbox"/>	<input type="checkbox"/>	13	F
En un árbol binario lleno, el camino mínimo de la raíz (longitud del camino más corto hasta un árbol vacío) es igual a la altura del árbol.	<input type="checkbox"/>	<input type="checkbox"/>	14	V
La altura máxima de un árbol de búsqueda digital es “ $n+1$ ”, siendo n el número de bits de la clave.	<input type="checkbox"/>	<input type="checkbox"/>	15	V

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED diciembre 2005

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
 - **El test vale un 40% de la nota de teoría.**
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F		
Las colas también se conocen como listas FIFO.	<input type="checkbox"/>	<input type="checkbox"/>	1.	V
Un árbol binario lleno es también un árbol 2-3-4.	<input type="checkbox"/>	<input type="checkbox"/>	2.	F
Sea un árbol 2-3-4 inicialmente vacío. Tras utilizar las operaciones de inserción y borrado de un árbol 2-3-4 siempre se consigue un árbol binario lleno.	<input type="checkbox"/>	<input type="checkbox"/>	3.	F
Un árbol binario completo es un árbol 2-3-4.	<input type="checkbox"/>	<input type="checkbox"/>	4.	F
Un árbol binario completo de altura h y con $2^h - 1$ nodos es un árbol 2-3-4.	<input type="checkbox"/>	<input type="checkbox"/>	5.	F
Un árbol Rojo – Negro con todos sus enlaces negros tiene los mismos nodos que el árbol 2-3-4 equivalente.	<input type="checkbox"/>	<input type="checkbox"/>	6.	V
En un árbol B todos los nodos han de tener al menos $m/2$ hijos o $(m-1)/2$ claves.	<input type="checkbox"/>	<input type="checkbox"/>	7.	F
En una tabla Hash con dispersión cerrada, las casillas vacías hay que diferenciarlas de las suprimidas para realizar una inserción.	<input type="checkbox"/>	<input type="checkbox"/>	8.	F
En un Hash cerrado, el factor de carga ($\alpha = n/B$) puede ser mayor que 1 cuando n sea mayor que B.	<input type="checkbox"/>	<input type="checkbox"/>	9.	F
En un multigrafo pueden existir infinitas aristas para un numero “n” de vértices.	<input type="checkbox"/>	<input type="checkbox"/>	10.	V
La operación concatena es un enriquecimiento de las operaciones definidas para el tipo cola.	<input type="checkbox"/>	<input type="checkbox"/>	11.	V
Es posible reconstruir un único árbol AVL a partir de un recorrido por niveles.	<input type="checkbox"/>	<input type="checkbox"/>	12.	V
El borrado en un árbol AVL nunca requiere más de una rotación en el camino de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	13.	F
En la escala de complejidades, la complejidad cuadrática es menor que la logarítmica.	<input type="checkbox"/>	<input type="checkbox"/>	14.	F

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED enero 2006

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **35 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		
En la especificación de un TAD, las operaciones auxiliares son visibles para los usuarios.	<input type="checkbox"/>	<input type="checkbox"/>	1.	F
Una operación del TAD X que tenga la sintaxis Crear() →X es una operación constructora generadora.	<input type="checkbox"/>	<input type="checkbox"/>	2.	V
En C++, los miembros <i>protected</i> son privados para el exterior, pero permiten el acceso a las clases derivadas.	<input type="checkbox"/>	<input type="checkbox"/>	3.	V
Dadas las clases <i>TDir</i> y <i>TVectorDir</i> : <pre>class TDir { public: private: int e1; char c1; }; class TVectorDir { public: private: TDir *vector; int dim; }; TVectorDir::~TVectorDir () { if (v!=NULL) delete v; dim=0; v=NULL; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	4.	F
¿Es correcta la implementación del destructor de <i>TVectorDir</i> ?	<input type="checkbox"/>	<input type="checkbox"/>	5.	F
El nivel de un nodo en un árbol coincide con la longitud del camino desde la raíz a dicho nodo	<input type="checkbox"/>	<input type="checkbox"/>	6.	V
Dado un único recorrido de un árbol binario lleno es posible reconstruir dicho árbol	<input type="checkbox"/>	<input type="checkbox"/>	7.	F
Sea el tipo <i>arbin</i> definido en clase. La semántica de la operación <i>nodos</i> es la siguiente: <pre>Var i,d:arbin; x:item; nodos(crear_arbin())=0 nodos(enraizar(i,x,d))=nodos(i)+nodos(d)</pre>	<input type="checkbox"/>	<input type="checkbox"/>	8.	F
A los árboles generales también se les llama árboles multicamino de búsqueda	<input type="checkbox"/>	<input type="checkbox"/>	9.	V
El ítem medio (según la relación de orden) almacenado en un árbol binario de búsqueda lleno siempre se encuentra en la raíz.	<input type="checkbox"/>	<input type="checkbox"/>	10.	V
Un árbol completo siempre está balanceado respecto a la altura	<input type="checkbox"/>	<input type="checkbox"/>	11.	F
En un árbol AVL siempre que se inserte una etiqueta hay que realizar una rotación	<input type="checkbox"/>	<input type="checkbox"/>	12.	V
El coste temporal en el peor caso de la operación de inserción en un árbol 2-3-4 es $\log_2(n+1) \approx \log_2(n)$ siendo “n” el número total de ítems	<input type="checkbox"/>	<input type="checkbox"/>	13.	V
Se puede obtener un único árbol 2-3-4 a partir de su recorrido por niveles	<input type="checkbox"/>	<input type="checkbox"/>	14.	V
La semántica de la operación <i>anterior</i> vista en clase es la siguiente: <pre>VAR L1: lista; x: item; p: posicion; anterior(L1, primera(L1)) = error_posicion(); si p != ultima(L1) entonces anterior(L1, siguiente(L1, p)) = p anterior(inscabeza(L1, x), primera(L1)) = primera(inscabeza(L1, x))</pre>	<input type="checkbox"/>	<input type="checkbox"/>	15.	F
Sea el tipo <i>vector</i> definido en clase. La semántica de la operación <i>recu</i> es la siguiente: <pre>Var v:vector; i,j:int; x:item; recu(crear_vector(),i)=error_item() si i > j entonces recu(asig(v,i,x),j)=recu(v,j) sino recu(asig(v,i,x),j)=TRUE</pre>	<input type="checkbox"/>	<input type="checkbox"/>	16.	V
<i>crear_pila()</i> , <i>apilar(pila,item)</i> y <i>desapilar(pila)</i> son operaciones constructoras del tipo <i>pila</i> .	<input type="checkbox"/>	<input type="checkbox"/>	17.	F
Todo árbol binario de búsqueda es un árbol 2-3-4	<input type="checkbox"/>	<input type="checkbox"/>	18.	F
La semántica de la operación <i>Base</i> que actúa sobre una <i>pila</i> y devuelve el primer elemento apilado es la siguiente: <pre>Base(crear_pila ()) = crear_pila () Base(apilar(crear_pila (), x)) = x Base(apilar(p, x)) = Base(p)</pre>	<input type="checkbox"/>	<input type="checkbox"/>		

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED febrero 2007

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **35 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		
En C++ y cuando se emplea composición (<i>layering</i>), los métodos de la clase derivada pueden acceder a la parte pública de la clase base.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
En C++, el puntero <i>this</i> no se puede emplear para modificar el objeto al que apunta.	<input type="checkbox"/>	<input type="checkbox"/>	2	F
En C++, los constructores se pueden invocar explícitamente cuando el programador lo desee (por ejemplo: <i>TLista a; a.TLista();</i>).	<input type="checkbox"/>	<input type="checkbox"/>	3	F
En C++, la siguiente declaración es incorrecta: $int\& a = 1;$	<input type="checkbox"/>	<input type="checkbox"/>	4	V
En la escala de complejidades se cumple que $O(\log n) \subset O(\log \log n)$.	<input type="checkbox"/>	<input type="checkbox"/>	5	F
El algoritmo de búsqueda binaria estudiado en clase (búsqueda de un elemento en un vector ordenado) tiene una complejidad de $\mathcal{O}(1)$.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
Para un vector de naturales, se define la operación <i>eliminar</i> que borra las posiciones pares del vector marcándolas con "0" (para calcular el resto de una división, se puede utilizar la operación MOD). La sintaxis y la semántica de la operación <i>eliminar</i> es la siguiente: $eliminar: vector \rightarrow vector$ <i>Var</i> v:vector; i: entero; x:natural; $eliminar(crear_vector()) = crear_vector()$ $si (i \text{ MOD } 2) == 0$ $entonces eliminar(asignar(v,i,x)) = asignar(eliminar(v),i,0)$ $si no eliminar(asignar(v,i,x)) = asignar(eliminar(v),i,x)$	<input type="checkbox"/>	<input type="checkbox"/>	7	V
La semántica de la operación <i>ultima</i> vista en clase es la siguiente: <i>VAR</i> L1: lista; x: item; <i>si</i> esvacía(L1) <i>entonces</i> $ultima(inscabeza(L1, x) = primera(L1)$ <i>si no</i> $ultima(inscabeza(L1, x) = ultima(L1)$	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Es posible reconstruir un único árbol binario de altura 6 a partir de un recorrido en preorden con 62 etiquetas.	<input type="checkbox"/>	<input type="checkbox"/>	9	F
La sintaxis y semántica de la operación <i>quita_hojas</i> , que actúa sobre un árbol binario y devuelve el árbol binario original sin sus hojas, es la siguiente: $quita_hojas(arbin) \rightarrow arbin$ <i>VAR</i> i, d: arbin; x: item; $quita_hojas(crea_arbin()) = crea_arbin()$ $quita_hojas(enraizar(crea_arbin(), x, crea_arbin())) = crea_arbin()$ $quita_hojas(enraizar(i, x, d)) =$ $enraizar(quita_hojas(i), x, quita_hojas(d))$	<input type="checkbox"/>	<input type="checkbox"/>	10	V
Dados los recorridos de preorden, postorden y niveles de un árbol binario de altura 2 y 1 hoja es posible reconstruir 2 árboles binarios.	<input type="checkbox"/>	<input type="checkbox"/>	11	V
Todo árbol AVL es un árbol 2-3-4	<input type="checkbox"/>	<input type="checkbox"/>	12	F
La operación (<i>DIVIDEHIJODE2</i> (p, q)) en la inserción de un elemento en un árbol 2-3-4 puede aumentar la altura del árbol original.	<input type="checkbox"/>	<input type="checkbox"/>	13	F
En el algoritmo del borrado de un elemento en un árbol 2-3-4 si q es 2-nodo y r es 3-nodo hay que hacer una ROTACIÓN.	<input type="checkbox"/>	<input type="checkbox"/>	14	V

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED febrero 2008

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **30 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		
La complejidad logarítmica aparece en algoritmos que descartan muchos valores (generalmente la mitad) en un único paso.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
Dada la sintaxis de la función $IC(lista, item) \rightarrow lista$, que inserta un elemento a la cabeza de la lista pasada como parámetro y $crear() \rightarrow lista$, que crea una lista vacía. La siguiente secuencia: $IC(IC(IC(crear(), a), b), c)$, daría como resultado una lista con los elementos en este orden: $a \rightarrow b \rightarrow c$, donde a es el primer elemento de la lista	<input type="checkbox"/>	<input type="checkbox"/>	2	F
Dentro de la especificación algebraica de los números naturales definimos la sintaxis de la función F como: $F: natural \rightarrow BOOL$, y su semántica como: $F(cero)=TRUE$, $F(suc(cero))=FALSE$, $F(suc(suc(x)))=F(x)$. Para el número natural $x=35$, la función F devolvería $TRUE$.	<input type="checkbox"/>	<input type="checkbox"/>	3	F
En C++, si un objeto se sale de ámbito entonces se invoca automáticamente al destructor de ese objeto.	<input type="checkbox"/>	<input type="checkbox"/>	4	V
En C++, el constructor de copia recibe como argumento un objeto del mismo tipo pasado por referencia o por valor.	<input type="checkbox"/>	<input type="checkbox"/>	5	F
El algoritmo de búsqueda binaria estudiado en clase (búsqueda de un elemento en un vector ordenado) tiene una complejidad de $\Omega(1)$.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
La complejidad espacial es la cantidad de recursos espaciales que un algoritmo consume o necesita para su ejecución	<input type="checkbox"/>	<input type="checkbox"/>	7	V
La operación <code>BorrarItem</code> , que borra todas las ocurrencias del item i que se encuentren en la lista, tiene la siguiente sintaxis y semántica: BorrarItem: LISTA, ITEM \rightarrow LISTA BorrarItem(Crear, i) = Crear BorrarItem(IC(L1, j), i) = si ($i == j$) entonces BorrarItem (L1, i) sino IC (BorrarItem (L1, i), j)	<input type="checkbox"/>	<input type="checkbox"/>	8	V
La semántica de la operación obtener en una lista con acceso por posición es la siguiente (IC= InsertarCabeza(Lista, Ítem), p: posición, l1: lista, x: ítem): obtener(crear(), p)=error_item() si p = primera(IC(l1, x)) entonces obtener(IC(l1, x), p)=x sino obtener(IC(l1, x), p)=IC(obtener(l1, p), x)	<input type="checkbox"/>	<input type="checkbox"/>	9	F
El máximo número de nodos en un nivel $i-1$ de un árbol binario es 2^{i-2} , $i \geq 2$	<input type="checkbox"/>	<input type="checkbox"/>	10	V
Sea el TIPO arbin definido en clase. La semántica de la operación nodos es la siguiente: Var i, d: arbin; x: item; nodos(crear_arbin())=0 nodos(enraizar(i, x, d))=nodos(i)+nodos(d)	<input type="checkbox"/>	<input type="checkbox"/>	11	F
El coste temporal en su peor caso de insertar una etiqueta en un árbol binario de búsqueda es lineal con la altura del árbol	<input type="checkbox"/>	<input type="checkbox"/>	12	V
Se puede obtener un único árbol 2-3-4 a partir de su recorrido por niveles	<input type="checkbox"/>	<input type="checkbox"/>	13	V

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED febrero 2009

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **25 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		
En C++, al declarar una clase "A" como AMIGA de otra clase "B", todas las funciones miembro de "B" automáticamente pasan a ser funciones AMIGAS de "A"	<input type="checkbox"/>	<input type="checkbox"/>	1	F
En C++, si una clase "B" se construye por composición (layering), a partir de otra clase "A", definiendo un objeto miembro de la clase "A" en su parte privada, al invocar al constructor de "B" se invoca antes al constructor de "A" y luego al de "B"	<input type="checkbox"/>	<input type="checkbox"/>	2	V
Las funciones y clases amigas se tienen que declarar en la parte pública de la clase.	<input type="checkbox"/>	<input type="checkbox"/>	3	F
Para el siguiente algoritmo, la complejidad temporal en su peor caso sería O(1): <pre>for (i=0; i<100; i++) for (j=0; j<100; j++) if (v[i]<v[j]) v[i]=v[j]; else v[j]=v[i];</pre>	<input type="checkbox"/>	<input type="checkbox"/>	4	V
La complejidad temporal en su peor caso del siguiente fragmento de código es O(n) <pre>int i, j, n, sum; for (i = 4; i < n; i++) { for (j = i-3, sum = a[i-4]; j <= i; j++) sum += a[j]; cout << "La suma del subarray " << i-4 << " es " << sum << endl; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	5	V
La semántica de la operación esvaciapos del tipo vector vista en clase es la siguiente: <pre>VAR v: vector; i, j: int; x: item; esvaciapos(crear(), i) = CIERTO esvaciapos(asig(v, i, x), j) si (i == j) entonces FALSO si no esvaciapos(v, j) fsi</pre>	<input type="checkbox"/>	<input type="checkbox"/>	6	V
La semántica de la operación anterior vista en clase es la siguiente: <pre>VAR L1: lista; x: item; p: posicion; anterior(L1, primera(L1)) = error_posicion(); si p != ultima(L1) entonces anterior(L1, siguiente(L1, p)) = p anterior(inscabeza(L1, x), primera(L1)) = L1</pre>	<input type="checkbox"/>	<input type="checkbox"/>	7	F
La sintaxis y semántica de la operación simétricos, que comprueba que 2 árboles binarios son simétricos, es la siguiente: <pre>simétricos(arbin, arbin) → bool VAR i1, d1, i2, d2: arbin; x, y: item; simétricos(enraizar(i1, x, d1), crea_arbin()) = FALSO simétricos(crea_arbin(), enraizar(i1, x, d1)) = FALSO simétricos(enraizar(i1, x, d1), enraizar(i2, y, d2)) = si (x == y) entonces (simétricos(i1, d2) & simétricos(d1, i2)) sino FALSO</pre>	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Un árbol binario completo es un AVL	<input type="checkbox"/>	<input type="checkbox"/>	9	F

Apellidos, Nombre:

DNI:

Examen PED enero 2010

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **15 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												
En C++, el valor de la variable q al finalizar este fragmento de código es 11:															
<pre>int q = 0; int k = 5; do { q += k; k++; } while(q < 7);</pre>															
La complejidad temporal (en su caso mejor) del siguiente fragmento de código es $\Omega(n)$															
<pre>int i, length, n, i1, i2, k; for (i = 0, length = 1; i < n-1; i++) { for (i1 = i2 = k = i; k < n-1 && a[k] < a[k+1]; k++, i2++); if (length < i2 - i1 + 1) length = i2 - i1 + 1; }</pre>															
La semántica de la operación insertar del tipo lista vista en clase es la siguiente:															
<pre>VAR L1: lista; x,y: item; p: posicion; insertar(crear(), p, x) = crear() si p == primera(inscabeza(L1, x)) entonces insertar(inscabeza(L1, x), p, y) = inscabeza(inscabeza(L1, x), y) si no insertar(inscabeza(L1, x), p, y) = inscabeza(insertar(L1, p, y), x)</pre>															
El grado de un árbol es el grado mínimo de todos los nodos de ese árbol															
El siguiente árbol es binario de búsqueda															
<pre>graph TD g((g)) --> b((b)) g --> m((m)) m --> j((j)) m --> p((p)) j --> f((f)) j --> k((k))</pre>															
Dada la siguiente representación secuencial del árbol binario A, <table border="1"><tr><td>1</td><td>4</td><td>8</td><td>1</td><td>9</td><td>5</td><td></td><td></td><td></td><td></td></tr></table> el elemento 5 es el hijo izquierda del elemento 8						1	4	8	1	9	5				
1	4	8	1	9	5										
En el algoritmo de borrado de un elemento de un árbol AVL, tenemos que actualizar los factores de equilibrio de todos los nodos que han intervenido en la búsqueda del elemento a borrar															
En el algoritmo del borrado de un elemento en un árbol 2-3-4 siempre que q sea 2-nodo hay que hacer una reestructuración.															
El árbol 2-3-4 no vacío tiene como mínimo dos claves en cada nodo															
La operación <i>BorrarItem</i> , que borra todas las ocurrencias del item i que se encuentren en la lista, tiene la siguiente sintaxis y semántica:															
<pre>BorrarItem: LISTA, ITEM -> LISTA BorrarItem(Crear, i) = Crear BorrarItem(IC(L1,j), i) = si (i == j) entonces BorrarItem (L1, i) sino IC (BorrarItem (L1, i), j)</pre>															

DNI:

Normas:

- La entrega del test no corre convocatoria.
- Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
- El test vale un 40% de la nota de teoría: **4 puntos**.
- En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F		
Dentro de la especificación algebraica de los números naturales definimos la sintaxis de la función F como: $F: \text{natural} \rightarrow \text{BOOL}$, y su semántica como: $F(\text{cero}) = \text{TRUE}$, $F(\text{suc}(\text{cero})) = \text{FALSE}$, $F(\text{suc}(\text{suc}(x))) = F(x)$. Para el número natural $x=35$, la función F devolvería FALSE.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
En C++, la memoria que se reserva con new se libera automáticamente por el destructor.	<input type="checkbox"/>	<input type="checkbox"/>	2	F
La mejor complejidad temporal que se puede conseguir en un algoritmo es $O(n)$, con “n” como la talla del problema.	<input type="checkbox"/>	<input type="checkbox"/>	3	F
La semántica de la operación <i>sublista</i> del tipo lista vista en clase es la siguiente: <i>VAR L: lista; x, y: item; n: natural; p: posicion;</i> <i>sublista(L, p, 0) = crear()</i> <i>sublista(crear(), p, n) = crear()</i> <i>si p == primera(inscabeza(L, x)) entonces</i> <i>sublista(inscabeza(L, x), p, n) = inscabeza(sublista(L, primera(L), n), x)</i> <i>si no sublista(inscabeza(L, x), p, n) = sublista(L, p, n)</i>	<input type="checkbox"/>	<input type="checkbox"/>	4	F
Dado un único recorrido de un árbol binario lleno, es posible reconstruir dicho árbol	<input type="checkbox"/>	<input type="checkbox"/>	5	V
Cuando realizamos un recorrido en inorden en un árbol binario de búsqueda las etiquetas aparecen ordenadas de menor a mayor	<input type="checkbox"/>	<input type="checkbox"/>	6	V
Todo árbol binario de búsqueda es un árbol mínimo	<input type="checkbox"/>	<input type="checkbox"/>	7	F
El siguiente árbol está balanceado con respecto a la altura	<input type="checkbox"/>	<input type="checkbox"/>	8	V
Todo árbol binario de búsqueda es un árbol 2-3.	<input type="checkbox"/>	<input type="checkbox"/>	9	F
En un árbol 2-3-4 sólo los nodos hoja y la raíz pueden ser de tipo 2-nodo	<input type="checkbox"/>	<input type="checkbox"/>	10	F
En un árbol rojo-negro la búsqueda de una etiqueta dependerá de los colores de los hijos de cada nodo	<input type="checkbox"/>	<input type="checkbox"/>	11	F
El árbol 2-3 es un árbol B m-camino de búsqueda con $m=2$	<input type="checkbox"/>	<input type="checkbox"/>	12	F
El TAD Diccionario es un subtipo del TAD Conjunto	<input type="checkbox"/>	<input type="checkbox"/>	13	V
La dispersión abierta elimina el problema del clustering secundario.	<input type="checkbox"/>	<input type="checkbox"/>	14	V
En un montículo doble de altura h se pueden almacenar un máximo de 2^{h-2} claves.	<input type="checkbox"/>	<input type="checkbox"/>	15	F
Un árbol leftist mínimo es un árbol binario mínimo tal que si no es vacío: $CMÍN(HijoIzq(x)) > CMÍN(HijoDer(x))$ para todo x no vacío	<input type="checkbox"/>	<input type="checkbox"/>	16	F

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED enero 2006

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **35 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		
En la especificación de un TAD, las operaciones auxiliares son visibles para los usuarios.	<input type="checkbox"/>	<input type="checkbox"/>	1.	F
Una operación del TAD X que tenga la sintaxis Crear() →X es una operación constructora generadora.	<input type="checkbox"/>	<input type="checkbox"/>	2.	V
En C++, los miembros <i>protected</i> son privados para el exterior, pero permiten el acceso a las clases derivadas.	<input type="checkbox"/>	<input type="checkbox"/>	3.	V
Dadas las clases <i>TDir</i> y <i>TVectorDir</i> : <pre>class TDir { public: private: int e1; char c1; }; class TVectorDir { public: private: TDir *vector; int dim; }; TVectorDir::~TVectorDir () { if (v!=NULL) delete v; dim=0; v=NULL; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	4.	F
¿Es correcta la implementación del destructor de <i>TVectorDir</i> ?	<input type="checkbox"/>	<input type="checkbox"/>	5.	F
El nivel de un nodo en un árbol coincide con la longitud del camino desde la raíz a dicho nodo	<input type="checkbox"/>	<input type="checkbox"/>	6.	V
Dado un único recorrido de un árbol binario lleno es posible reconstruir dicho árbol	<input type="checkbox"/>	<input type="checkbox"/>	7.	F
Sea el tipo <i>arbin</i> definido en clase. La semántica de la operación <i>nodos</i> es la siguiente: <pre>Var i,d:arbin; x:item; nodos(crear_arbin())=0 nodos(enraizar(i,x,d))=nodos(i)+nodos(d)</pre>	<input type="checkbox"/>	<input type="checkbox"/>	8.	F
A los árboles generales también se les llama árboles multicamino de búsqueda	<input type="checkbox"/>	<input type="checkbox"/>	9.	V
El ítem medio (según la relación de orden) almacenado en un árbol binario de búsqueda lleno siempre se encuentra en la raíz.	<input type="checkbox"/>	<input type="checkbox"/>	10.	V
Un árbol completo siempre está balanceado respecto a la altura	<input type="checkbox"/>	<input type="checkbox"/>	11.	F
En un árbol AVL siempre que se inserte una etiqueta hay que realizar una rotación	<input type="checkbox"/>	<input type="checkbox"/>	12.	V
El coste temporal en el peor caso de la operación de inserción en un árbol 2-3-4 es $\log_2(n+1) \approx \log_2(n)$ siendo “n” el número total de ítems	<input type="checkbox"/>	<input type="checkbox"/>	13.	V
Se puede obtener un único árbol 2-3-4 a partir de su recorrido por niveles	<input type="checkbox"/>	<input type="checkbox"/>	14.	V
La semántica de la operación <i>anterior</i> vista en clase es la siguiente: <pre>VAR L1: lista; x: item; p: posicion; anterior(L1, primera(L1)) = error_posicion(); si p != ultima(L1) entonces anterior(L1, siguiente(L1, p)) = p anterior(inscabeza(L1, x), primera(L1)) = primera(inscabeza(L1, x))</pre>	<input type="checkbox"/>	<input type="checkbox"/>	15.	F
Sea el tipo <i>vector</i> definido en clase. La semántica de la operación <i>recu</i> es la siguiente: <pre>Var v:vector; i,j:int; x:item; recu(crear_vector(),i)=error_item() si i > j entonces recu(asig(v,i,x),j)=recu(v,j) sino recu(asig(v,i,x),j)=TRUE</pre>	<input type="checkbox"/>	<input type="checkbox"/>	16.	V
<i>crear_pila()</i> , <i>apilar(pila,item)</i> y <i>desapilar(pila)</i> son operaciones constructoras del tipo <i>pila</i> .	<input type="checkbox"/>	<input type="checkbox"/>	17.	F
Todo árbol binario de búsqueda es un árbol 2-3-4	<input type="checkbox"/>	<input type="checkbox"/>	18.	F
La semántica de la operación <i>Base</i> que actúa sobre una <i>pila</i> y devuelve el primer elemento apilado es la siguiente: <pre>Base(crear_pila ()) = crear_pila () Base(apilar(crear_pila (), x)) = x Base(apilar(p, x)) = Base(p)</pre>	<input type="checkbox"/>	<input type="checkbox"/>		

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen TAD/PED septiembre 2004

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
 - **El test vale un 40% de la nota de teoría.**
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F	
Las operaciones modificadoras de un TAD permiten generar, por aplicaciones sucesivas, todos los valores del TAD a especificar.	<input type="checkbox"/>	<input type="checkbox"/>	F 1.
Sea el método Primera perteneciente a la clase Tlista que devuelve la primera posición de la lista que lo invoca: <pre>TPosicion Tlista::Primera() class Tlista { { TPosicion p; public: ... p.pos = lis; private: return p; } Tnodo *lis; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	F 2.
En el método Primera, se invoca al constructor de Tlista.			
Sea el tipo cola definido en clase. La semántica de la operación cabeza es la siguiente: Var c:cola; x:item; cabeza(crear_cola())=error_item() si esvacía(c) entonces cabeza(encolar(c,x))=x sino cabeza(encolar(c,x))=encolar(cabeza(c),x)	<input type="checkbox"/>	<input type="checkbox"/>	F 3.
El recorrido en postorden de un árbol binario es el inverso especular del recorrido en preorden del mismo árbol	<input type="checkbox"/>	<input type="checkbox"/>	V 4.
En la operación de borrado de un ítem en un árbol AVL, si se realiza una rotación II, al menos es necesario realizar otra rotación de cualquier tipo.	<input type="checkbox"/>	<input type="checkbox"/>	F 5.
Los nodos de grado 0 de un árbol 2-3 han de estar en el mismo nivel del árbol	<input type="checkbox"/>	<input type="checkbox"/>	V 6.
Al insertar un elemento en un árbol 2-3-4 se pueden realizar una operación de DIVIDERAIZ y otra de DIVIDEHIJODE2.	<input type="checkbox"/>	<input type="checkbox"/>	V 7.
En un árbol rojo-negro ha de haber al menos un enlace rojo.	<input type="checkbox"/>	<input type="checkbox"/>	F 8.
En un árbol B tiene que haber el mismo número de nodos en el hijo izquierdo de la raíz que en el hijo derecho.	<input type="checkbox"/>	<input type="checkbox"/>	F 9.
La especificación algebraica de la siguiente operación indica que se devolverá el número de elementos del conjunto multiplicado por 3 (C: Conjunto; x: Ítem): Operación(Crear) \Leftrightarrow 0 Operación (Insertar(C, x)) \Leftrightarrow 3 + Operación(C)	<input type="checkbox"/>	<input type="checkbox"/>	V 10.
En la dispersión abierta se pueden producir colisiones entre claves sinónimas y no sinónimas	<input type="checkbox"/>	<input type="checkbox"/>	F 11.
Un recorrido en inorden de un montículo nos devolverá todos los elementos de forma ordenada.	<input type="checkbox"/>	<input type="checkbox"/>	F 12.
Un árbol 2-3 cumple las propiedades de un árbol Leftist.	<input type="checkbox"/>	<input type="checkbox"/>	F 13.
Un bosque extendido en profundidad de un grafo dirigido al que se le añaden los arcos de cruce es un grafo acíclico dirigido.	<input type="checkbox"/>	<input type="checkbox"/>	V 14.

Apellidos:

Nombre:

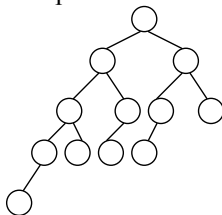
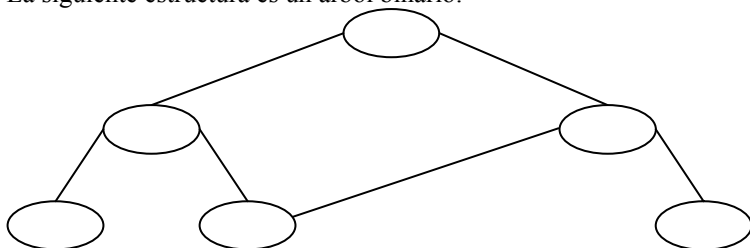
Convocatoria:

DNI:

Examen TAD/PED septiembre 2005

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
 - **El test vale un 40% de la nota de teoría.**
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	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

Apellidos:

Nombre:

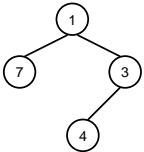
Convocatoria:

DNI:

Examen TAD/PED septiembre 2006

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **18 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. A continuación comenzará el siguiente ejercicio.**
 - El test vale un 40% de la nota de teoría.
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F	
Las operaciones constructoras generadoras de un tipo permiten obtener cualquier valor de dicho tipo.	<input type="checkbox"/>	<input type="checkbox"/>	1. V
En C++, si no se ha implementado la sobrecarga del operador asignación, se invoca automáticamente al constructor de copia.	<input type="checkbox"/>	<input type="checkbox"/>	2. F
Es posible reconstruir un único árbol binario de altura 6 a partir de un recorrido en postorden con 63 etiquetas.	<input type="checkbox"/>	<input type="checkbox"/>	3. V
La semántica de la operación nodos del tipo <i>arbin</i> vista en clase es la siguiente: $VAR i, d: arbin; x: item;$ $nodos(crea_arbin()) = 0$ $nodos(enraizar(i, x, d)) = nodos(i) + nodos(d)$	<input type="checkbox"/>	<input type="checkbox"/>	4. F
Se puede reconstruir un único árbol binario cualquiera teniendo sus recorridos en preorden y postorden.	<input type="checkbox"/>	<input type="checkbox"/>	5. F
La semántica de la operación <i>recu</i> vista en clase es la siguiente: $VAR v: vector; i, j: int; x: item;$ $recu(crear_vector(), i) = error_item()$ $recu(asig(v, i, x), j)$ si $(i == j)$ entonces x sino FALSO fsi	<input type="checkbox"/>	<input type="checkbox"/>	6. F
En un árbol AVL cuyo factor de equilibrio es -2, al insertar un elemento en la rama derecha, el árbol vuelve al estado de equilibrio.	<input type="checkbox"/>	<input type="checkbox"/>	7.
Dado un árbol 2-3 de altura h con n items con todos sus nodos del tipo 2-Nodo: la complejidad de la operación de búsqueda de un ítem es $O(\log_2 n)$	<input type="checkbox"/>	<input type="checkbox"/>	8. V
Un árbol binario de búsqueda lleno de altura 4 es un árbol 2-3-4, pero no se puede conseguir a partir de un árbol inicialmente vacío y utilizando las operaciones de inserción y borrado de un árbol 2-3-4	<input type="checkbox"/>	<input type="checkbox"/>	9. V
Un grafo no dirigido puede tener aristas que empiecen y acaben en el mismo vértice.	<input type="checkbox"/>	<input type="checkbox"/>	10. F
El siguiente árbol es leftist mínimo: 	<input type="checkbox"/>	<input type="checkbox"/>	11. V
Un trie cumple las propiedades de un árbol general.	<input type="checkbox"/>	<input type="checkbox"/>	12. V

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen TAD/PED julio 2004

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
 - **El test vale un 40% de la nota de teoría.**
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F						
La operación crear_pila() es constructora modificadora.	<input type="checkbox"/>	<input type="checkbox"/>	F 1.					
En C++, una forma correcta de copiar una cadena es la siguiente: <i>char a[50] = “Tipos Abstractos de Datos”;</i> <i>char *b;</i> <i>b = new char[strlen(a)];</i> <i>strcpy(b, a);</i>	<input type="checkbox"/>	<input type="checkbox"/>	F 2.					
El caso peor de la búsqueda es más eficiente en una lista ordenada que en una lista cuyos elementos no están ordenados.	<input type="checkbox"/>	<input type="checkbox"/>	F 3.					
En un árbol binario cada elemento puede tener como máximo dos predecesores.	<input type="checkbox"/>	<input type="checkbox"/>	F 4.					
Si se implementa el algoritmo de ordenación de un vector “heapsort” utilizando un heap máximo los elementos quedan ordenados en el vector de forma descendente.	<input type="checkbox"/>	<input type="checkbox"/>	F 5.					
Dado un recorrido en preorden (RID) de un árbol AVL es posible reconstruir un único árbol AVL.	<input type="checkbox"/>	<input type="checkbox"/>	V 6.					
El número máximo de elementos que se puede almacenar en un árbol 2-3 de altura h es 3 ^h -1	<input type="checkbox"/>	<input type="checkbox"/>	V 7.					
Al borrar un elemento en un árbol 2-3-4 se puede realizar una operación de DIVIDERAIZ.	<input type="checkbox"/>	<input type="checkbox"/>	F 8.					
En un árbol B m-camino de búsqueda con m=16: en cualquier nodo excepto la raíz hay 8 ítems como mínimo.	<input type="checkbox"/>	<input type="checkbox"/>	F 9.					
La especificación algebraica de la siguiente operación permite la inserción de claves repetidas (C: ConjuntoConClavesRepetidas; x, y: Ítem): Insertar(Insertar(C, x), y) ⇔ Insertar(Insertar(C,y), x)	<input type="checkbox"/>	<input type="checkbox"/>	V 10.					
En el TAD Diccionario con dispersión cerrada, con función de redispersión “h _i (x)=(H(x) + k(x)*i) MOD B”, con B=6 se puede dar la situación de que en una búsqueda no se acceda a todas las posiciones de la tabla.	<input type="checkbox"/>	<input type="checkbox"/>	V 11.					
El siguiente vector representa un montículo máximo: <table border="1"><tr><td>10</td><td>5</td><td>3</td><td>1</td><td>2</td></tr></table>	10	5	3	1	2	<input type="checkbox"/>	<input type="checkbox"/>	V 12.
10	5	3	1	2				
Para todo nodo de un árbol Leftist, se cumple que la altura de su hijo izquierdo es menor que la de su hijo derecho.	<input type="checkbox"/>	<input type="checkbox"/>	F 13.					
La complejidad temporal de la búsqueda en un trie es lineal respecto al número de palabras almacenadas	<input type="checkbox"/>	<input type="checkbox"/>	F 14.					
Un bosque extendido en profundidad de un grafo no dirigido es un grafo acíclico.	<input type="checkbox"/>	<input type="checkbox"/>	V 15.					

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen TAD/PED Junio 2005

Modalidad 0

Normas:

- La entrega del test no corre convocatoria.
- Tiempo para efectuar el test: **20 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. A continuación comenzará el siguiente ejercicio.**
- **El test vale un 40% de la nota de teoría.**
- En la **HOJA DE CONTESTACIONES** el verdadero se corresponderá con la **A** y el falso con la **B**.

	V	F	
Siempre que se realiza una rotación DD en el borrado de un elemento en un árbol AVL decrece la altura del subárbol sobre el que se realiza la rotación.	<input type="checkbox"/>	<input type="checkbox"/>	1 F
Un árbol 2-3 es un árbol m -camino de búsqueda con $m=3$.	<input type="checkbox"/>	<input type="checkbox"/>	2 V
La complejidad temporal (en su peor caso) de buscar un elemento en un vector ordenado utilizando un algoritmo de búsqueda binaria es $O(\log_n 2)$ [siendo n el número de elementos].	<input type="checkbox"/>	<input type="checkbox"/>	3 F
Un árbol rojo-negro, en el que no hay ningún enlace rojo, es un árbol binario lleno	<input type="checkbox"/>	<input type="checkbox"/>	4 V
En una tabla de dispersión cerrada con la siguiente función de redispersión para la clave 14: $h_i(14) = (28 + 7 \cdot i) \text{ MOD } 2000$, se recorrerán todas las posiciones de la tabla buscando una posición libre.	<input type="checkbox"/>	<input type="checkbox"/>	5 V
En el algoritmo HeapSort, después del primer paso de insertar todos los elementos en un Heap representado como un vector, los elementos del mismo quedan ordenados.	<input type="checkbox"/>	<input type="checkbox"/>	6 F
Un bosque extendido en profundidad de un grafo dirigido al que se le añaden los arcos de avance y de cruce es un grafo acíclico dirigido.	<input type="checkbox"/>	<input type="checkbox"/>	7 V
Sea $G=(V,A)$ un grafo dirigido. Diremos que $G''=(V'',A'')$ es un árbol extendido de $G \Leftrightarrow V''=V, A'' \subset A, \forall v \in V'' \Rightarrow \text{grado}_E(v) \leq 1$.	<input type="checkbox"/>	<input type="checkbox"/>	8 V
La operación <i>BorrarItem</i> , que borra la primera ocurrencia del item i que se encuentre en la lista, tiene la siguiente sintaxis y semántica: BorrarItem: LISTA, ITEM \rightarrow LISTA BorrarItem(Crear, i) = Crear BorrarItem(IC(L1, j), i) = si ($i == j$) entonces BorrarItem (L1, i) sino IC (BorrarItem (L1, i), j)	<input type="checkbox"/>	<input type="checkbox"/>	9 F
Se puede reconstruir un único árbol binario de búsqueda teniendo sus recorridos en preorden y postorden.	<input type="checkbox"/>	<input type="checkbox"/>	10 V
La estructura de datos árbol aparece porque los elementos que lo constituyen mantienen una estructura jerárquica, obtenida a partir de estructuras lineales, al eliminar el requisito de que cada elemento tiene como máximo un predecesor.	<input type="checkbox"/>	<input type="checkbox"/>	11 F
Al realizar un recorrido en inorden de un montículo obtenemos una sucesión de claves ordenadas.	<input type="checkbox"/>	<input type="checkbox"/>	12 F
La función de redispersión en una tabla hash abierta tiene que cumplir como condición para que se recorran todas las posiciones del vector que el valor de B sea primo	<input type="checkbox"/>	<input type="checkbox"/>	13 F
Un grafo que tiene componentes fuertemente conexas es un grafo necesariamente libre de ciclos	<input type="checkbox"/>	<input type="checkbox"/>	14 F

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED junio 2008

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **15 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		
Las ecuaciones (vistas en clase) que permiten realizar la multiplicación de números naturales son las siguientes: $\begin{aligned} &VAR\ x, y: natural; \\ &mult(cero, x) = cero \\ &mult(x, cero) = cero \\ &mult(suc(y), x) = suma(mult(y, x), x) \end{aligned}$	<input type="checkbox"/>	<input type="checkbox"/>	1	V
En C++ y cuando se emplea composición (layering), los métodos de la clase derivada pueden acceder a la parte protegida de la clase base.	<input type="checkbox"/>	<input type="checkbox"/>	2	F
Para el siguiente algoritmo, la complejidad sería $O(n^2)$: $\begin{aligned} &for\ (i=0; i<100; i++) \\ &\quad for\ (j=0; j<100; j++) \\ &\quad\quad if\ (v[i]<v[j])\ \ v[i]=v[j]; \\ &\quad\quad else\ v[j]=v[i]; \end{aligned}$	<input type="checkbox"/>	<input type="checkbox"/>	3	F
Dados los recorridos de preorden, postorden y niveles de un árbol binario sólo se puede reconstruir un único árbol binario.	<input type="checkbox"/>	<input type="checkbox"/>	4	F
Sabiendo que A es un árbol binario de búsqueda completo y dado su recorrido inorden 1,4,6,7,9,12,14,20,23. La secuencia 12,7,20,4,9,14,23,1,6, se corresponde con su recorrido por niveles.	<input type="checkbox"/>	<input type="checkbox"/>	5	V
Cuando se borra un elemento en un AVL habrá casos en los que no sea necesario reestructurar el árbol. Para aquellos en los que sí se necesita reestructuración, se realizará una única rotación.	<input type="checkbox"/>	<input type="checkbox"/>	6	F
Dado un árbol 2-3, si la clave a borrar x está en un nodo no hoja, x se podría sustituir por la clave anterior a x en el recorrido en inorden del árbol, y continuar con el algoritmo de borrado.	<input type="checkbox"/>	<input type="checkbox"/>	7	V
La altura h de un árbol 2-3-4 con n elementos se encuentra entre los límites $\log_4 (n+1) \geq h \geq \log_2 (n+1)$.	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Un árbol binario completo es un árbol 2-3-4	<input type="checkbox"/>	<input type="checkbox"/>	9	F
Un árbol Rojo – Negro de altura 2 puede tener todos sus enlaces de color rojo	<input type="checkbox"/>	<input type="checkbox"/>	10	V
En la inserción en un árbol B se realiza un recorrido descendente desde la raíz en el que se van dividiendo los nodos llenos que nos encontremos por el camino hasta las hojas.	<input type="checkbox"/>	<input type="checkbox"/>	11	F

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED junio 2010

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **20 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		
En C++, la expresión <code>return *c;</code> devuelve la dirección de memoria de la variable <code>c</code> .	<input type="checkbox"/>	<input type="checkbox"/>	1	F
En un multigrafo pueden existir infinitas aristas para un número "n" de vértices.	<input type="checkbox"/>	<input type="checkbox"/>	2	V
La semántica de la operación obtener del tipo lista vista en clase es la siguiente: VAR L1: lista; x: item; p: posicion; obtener(crear(), p) = error_item() si p == primera(inscabeza(L1, x)) entonces obtener(inscabeza(L1, x), p) = x si no obtener(inscabeza(L1, x), p) = inscabeza(obtener(L1, p), x)	<input type="checkbox"/>	<input type="checkbox"/>	3	F
El nivel de un nodo en un árbol coincide con la longitud del camino desde la raíz a dicho nodo	<input type="checkbox"/>	<input type="checkbox"/>	4	F
A los árboles generales también se les llama árboles multicamino de búsqueda	<input type="checkbox"/>	<input type="checkbox"/>	5	F
Un árbol binario de búsqueda equilibrado respecto a la altura tiene una complejidad temporal en su peor caso en la búsqueda de $O(\log_2(n))$, con n el número de elementos del árbol.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
En un árbol 2-3 la altura siempre disminuye si al borrar un elemento se produce una combinación con los elementos de la raíz del árbol	<input type="checkbox"/>	<input type="checkbox"/>	7	F
En la operación de borrado de un elemento en un árbol 2-3-4, si hay que realizar reestructuraciones, éstas se realizan desde las hojas hacia la raíz.	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Las rotaciones en un árbol Rojo – Negro requieren un cambio de color en los nodos implicados.	<input type="checkbox"/>	<input type="checkbox"/>	9	F
Todo árbol binario de búsqueda es un árbol B con $m=3$.	<input type="checkbox"/>	<input type="checkbox"/>	10	F
En la dispersión cerrada puede haber colisiones entre claves sinónimas y no sinónimas.	<input type="checkbox"/>	<input type="checkbox"/>	11	V
Un Heap Mínimo es un árbol binario que además es árbol mínimo	<input type="checkbox"/>	<input type="checkbox"/>	12	F
En un árbol leftist se cumple que: $CMÍN(x) = 1 + CMÍN(HijoDer(x))$ para todo x no vacío y x con dos hijos	<input type="checkbox"/>	<input type="checkbox"/>	13	V

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED junio 2006

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **15 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							
En C++, la instrucción <i>TPila a = b</i> ; donde <i>b</i> es de tipo <i>TPila</i> invoca al constructor de copia.	<input type="checkbox"/>	<input type="checkbox"/>	1.	V					
La semántica de la operación <i>concatena</i> del tipo cola vista en clase es la siguiente: VAR c, q: cola; x: item; concatena(c, crear_cola ()) = c concatena(crear_cola (), c) = c concatena(c, encolar(q, x)) = encolar(concatena(c, q), x)	<input type="checkbox"/>	<input type="checkbox"/>	2.	V					
Se puede reconstruir un árbol binario cualquiera teniendo sus recorridos en preorden e inorden.	<input type="checkbox"/>	<input type="checkbox"/>	3.	V					
En la operación de borrado de un elemento en un árbol AVL, si se realiza una rotación doble siempre decrece la altura del árbol sobre el que se ha realizado la rotación.	<input type="checkbox"/>	<input type="checkbox"/>	4.	V					
El número máximo de elementos que se puede almacenar en un árbol 2-3 de altura <i>h</i> coincide con el número de elementos que hay en un árbol binario lleno de altura <i>h</i> .	<input type="checkbox"/>	<input type="checkbox"/>	5.	F					
En un árbol 2-3-4, los nodos pueden tener 1, 2 ó 3 hijos.	<input type="checkbox"/>	<input type="checkbox"/>	6.	F					
Todo árbol Rojo – Negro es un árbol 2-3-4.	<input type="checkbox"/>	<input type="checkbox"/>	7.	F					
Todo árbol B con m=4 es un árbol 2-3	<input type="checkbox"/>	<input type="checkbox"/>	8.	F					
El siguiente vector representa un montículo máximo: <table border="1" data-bbox="113 1184 995 1249"><tr><td>10</td><td>5</td><td>(Vacío)</td><td>1</td><td>2</td></tr></table>	10	5	(Vacío)	1	2	<input type="checkbox"/>	<input type="checkbox"/>	9.	F
10	5	(Vacío)	1	2					
Un bosque extendido en profundidad de un grafo dirigido al que se le añaden los arcos de avance es un grafo acíclico dirigido.	<input type="checkbox"/>	<input type="checkbox"/>	10.	V					

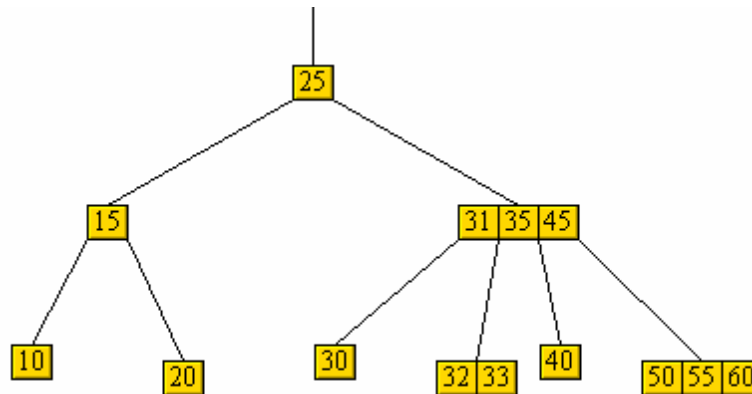
Examen PED junio 2006

- Normas:**
- Tiempo para efectuar el ejercicio: **2 horas**
 - En la cabecera de cada hoja **Y EN ESTE ORDEN** hay que poner: *Apellidos, Nombre*.
 - 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:** 3 de julio por la mañana. **Fecha de revisión de exámenes:** 3 de julio, 17 h. en aula SIN01 de la EPS IV
 - **Los alumnos que estén en 5ª o 6ª convocatoria deben indicarlo en la cabecera de todas las hojas**

1. Utilizando exclusivamente las operaciones constructoras generadoras del tipo grafo, definid la semántica de la operación 'examen' que se aplica sobre un grafo dirigido ponderado cuyos arcos están etiquetados con números naturales (es decir, el peso de los arcos son números naturales) y devuelve el número de arcos cuyo peso es igual a uno especificado. La sintaxis de la operación 'examen' es la siguiente: *examen: grafo, natural_peso → natural*

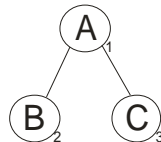
2. Dado el siguiente árbol 2-3-4, realiza las siguientes operaciones de forma sucesiva sobre el árbol obtenido en cada operación (criterio: sustituir por el mayor de la izquierda, elegir hijo izquierda cuando haya dos):

- Inserta 11, 12, 13, 65
- Borra 35, 30



3. Dada la siguiente expresión aritmética: $[(a+b)^2 + (b+c)] * [(d+f) / (d+e)]$.

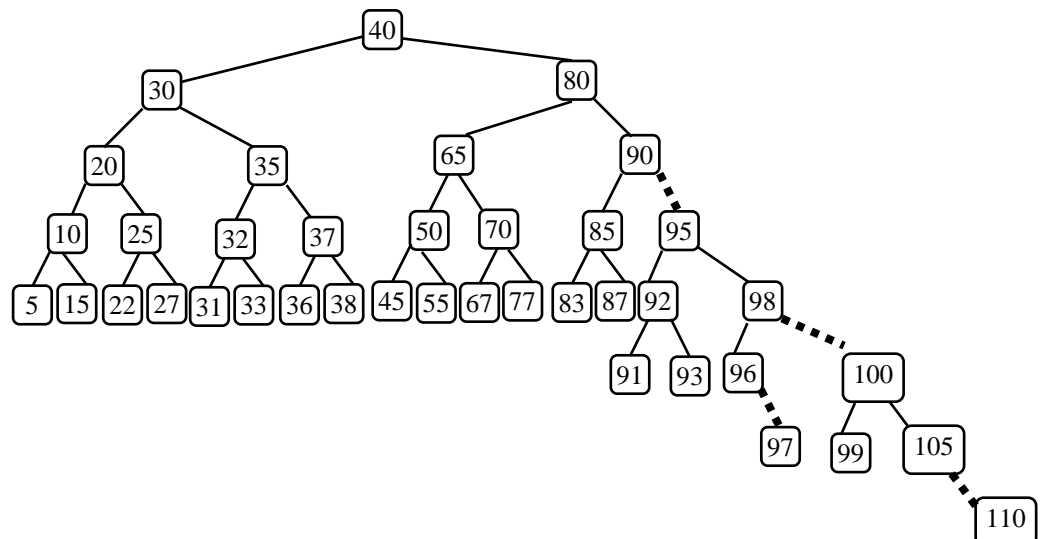
- Construir el GAD (grafo acíclico dirigido) que lo representa, utilizando exclusivamente los operadores + * /. Numera los nodos. Ejemplo:



- Calcula el bosque extendido en profundidad partiendo desde el único nodo que no tiene adyacencia de entrada. En el recorrido expresa además de la etiqueta del nodo, la numeración anterior. Recorrer la adyacencia de salida de un vértice continuando por el de menor numeración. Ejemplo: A_1, B_2, C_3
- Etiqueta los arcos.

4. a) Sobre el siguiente árbol R-N, realizar la inserción de las claves 130 y 120, detallando los cambios de color y rotaciones empleadas. **No será válido**, realizar la inserción como si fuese un árbol 2-3-4 y realizar una transformación final a R-N, en caso contrario no se puntuará la pregunta.

b) Obtener de forma razonada la complejidad en el peor caso de la operación de inserción en un árbol R-N.



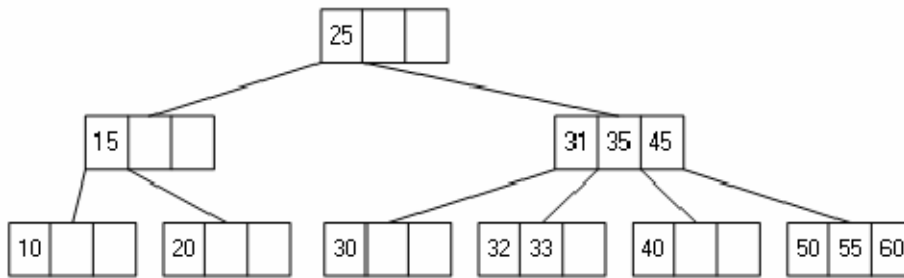
Examen PED junio 2006. Soluciones

1.

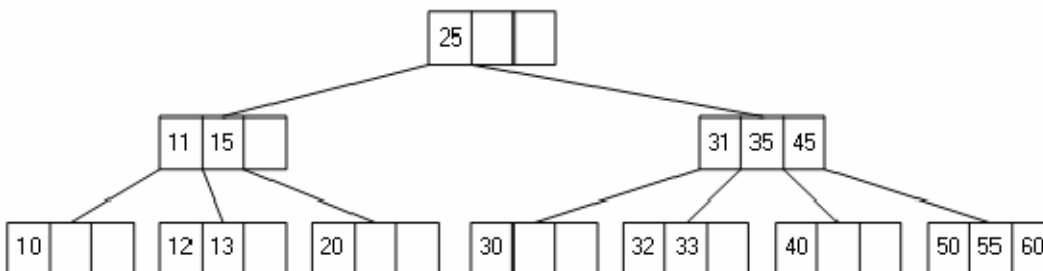
```

Var G: grafo; x,y: vértice; p,q: natural;
examen(crear_grafo(),q)=0
examen(InsertarArista(G,x,y,p),q)=
    si (q == p)
    entonces 1 + examen(G,q)
    si no examen(G,q)
    
```

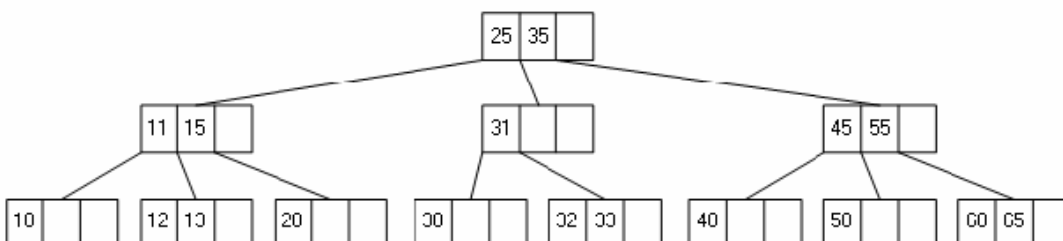
2.



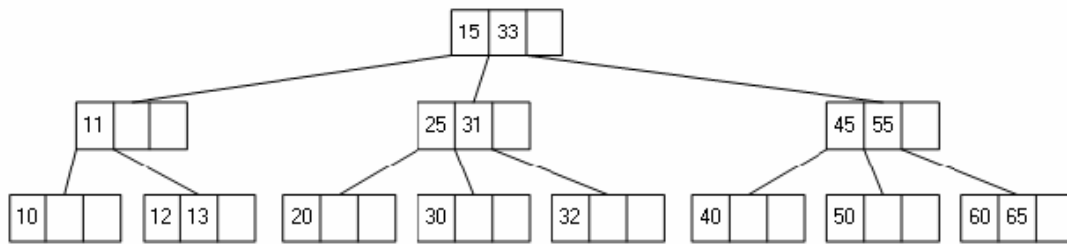
Inserta 11, 12, 13



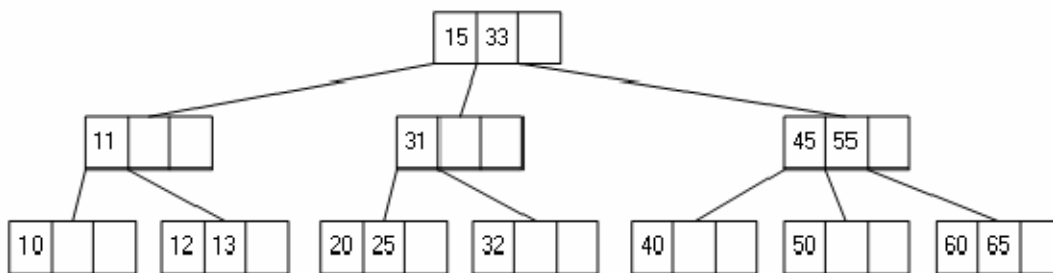
Inserta 65



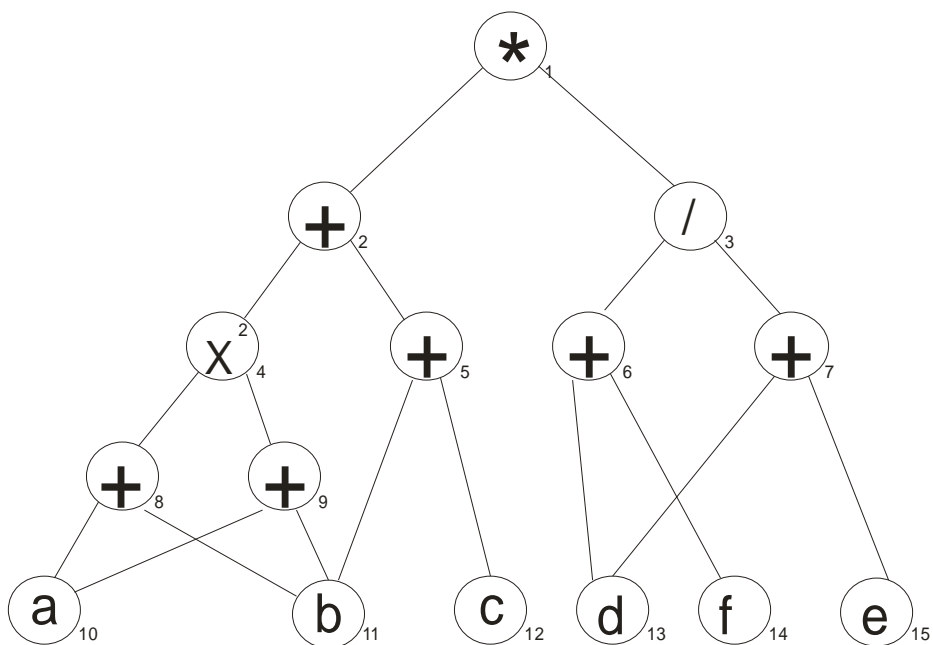
Borra 35



Borra 30



3. a)



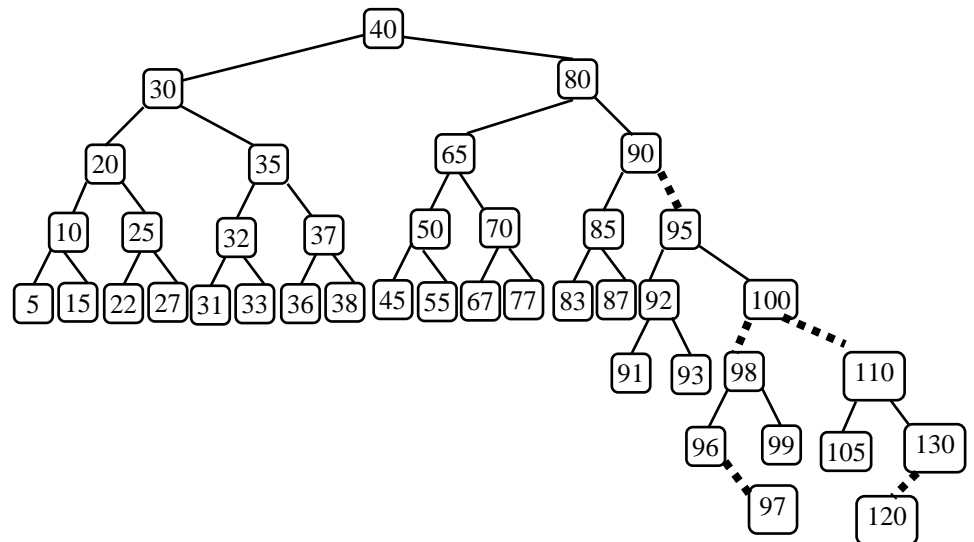
b)

Recorrido: $*_1$, $+_2$, x^2_4 , $+_8$, a_{10} , b_{11} , $+_9$, $+_5$, c_{12} , $/_3$, $+_6$, d_{13} , f_{14} , $+_7$, e_{15}

c)

(1,2)→Arbol
 (2,4) →Arbol
 (4,8)→Arbol
 (8,10)→Arbol
 (8,11)→ Arbol
 (4,9) →Arbol
 (9,10) → Cruce
 (9,11) → Cruce
 (2,5)→Arbol
 (5,11) → Cruce
 (5,12)→Arbol
 (1,3) →Arbol
 (3,6) → Arbol
 (6,13)→Arbol
 (6,14) →Arbol
 (3,7)→Arbol
 (7,13) →Cruce
 (7,15)→Arbol

4. a) Inserción del 130: rotación DD
 Inserción del
 120: cambio de color y rotación
 DD



b) $O(h)$, con h la altura del árbol, ya que la inserción en su peor caso comenzará en la raíz y finalizará en las hojas, sin recorrido de vuelta atrás.