

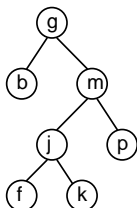
Apellidos, Nombre:

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>	<input type="checkbox"/>	<input type="checkbox"/>	1	V										
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>	<input type="checkbox"/>	<input type="checkbox"/>	2	V										
La semántica de la operación insertar del tipo lista vista en clase es la siguiente: VAR LI: lista; x,y: item; p: posicion; insertar(crear(), p, x) = crear() si p == primera(inscabeza(LI, x)) entonces insertar(inscabeza(LI, x), p, y) = inscabeza(inscabeza(LI, x), y) si no insertar(inscabeza(LI, x), p, y) = inscabeza(insertar(LI, p, y), x)	<input type="checkbox"/>	<input type="checkbox"/>	3	F										
El grado de un árbol es el grado mínimo de todos los nodos de ese árbol	<input type="checkbox"/>	<input type="checkbox"/>	4	F										
El siguiente árbol es binario de búsqueda 	<input type="checkbox"/>	<input type="checkbox"/>	5	F										
Dada la siguiente representación secuencial del árbol binario A, <table border="1" data-bbox="841 1590 1019 1619"><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					<input type="checkbox"/>	<input type="checkbox"/>	6	V
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	<input type="checkbox"/>	<input type="checkbox"/>	7	V										
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.	<input type="checkbox"/>	<input type="checkbox"/>	8	V										
El árbol 2-3-4 no vacío tiene como mínimo dos claves en cada nodo	<input type="checkbox"/>	<input type="checkbox"/>	9	F										
La operación <i>BorrarItem</i> , que borra todas las ocurrencias del item <i>i</i> que se encuentren en la lista, tiene la siguiente sintaxis y semántica: 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)	<input type="checkbox"/>	<input type="checkbox"/>	10	V										