

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 checked="" type="checkbox"/> 1 F
Un árbol 2-3 es un árbol m -camino de búsqueda con $m=3$.	<input checked="" 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 checked="" 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 checked="" 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 checked="" 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 checked="" 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 checked="" 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 checked="" type="checkbox"/> 9 F
Se puede reconstruir un único árbol binario de búsqueda teniendo sus recorridos en preorden y postorden.	<input checked="" 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 checked="" type="checkbox"/> 11 F
Al realizar un recorrido en inorden de un montículo obtenemos una sucesión de claves ordenadas.	<input type="checkbox"/>	<input checked="" 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 checked="" type="checkbox"/> 13 F
Un grafo que tiene componentes fuertemente conexas es un grafo necesariamente libre de ciclos	<input type="checkbox"/>	<input checked="" type="checkbox"/> 14 F

Examen TAD/PED junio 2005

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

- En la cabecera de cada hoja **Y EN ESTE ORDEN** hay que poner: *Apellidos, Nombre*. Cada pregunta se escribirá en folios diferentes.
- Se dispone de 20 minutos para abandonar el examen sin que corra convocatoria.
- Las soluciones al examen se dejarán en el campus virtual.
- Se puede escribir el examen con lápiz, siempre que sea legible
- **Todas las preguntas tienen el mismo valor.** Este examen vale el 60% de la nota de teoría.
- **Publicación de notas de exámenes:** 4-Julio. **Fecha de revisión de exámenes y de prácticas:** 8 de Julio. Hora de 15:00 a 17:00

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

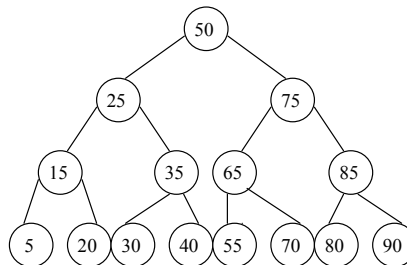
1. Utilizando exclusivamente las operaciones constructoras generadoras del vector, definid la sintaxis y la semántica de la operación ‘inversa’ que crea el vector inverso de un vector dado.

Nota: Se asume que el vector exclusivamente tiene 100 posiciones y el rango de las mismas es de 1...100, es de números naturales, y para éstos están definidas las operaciones de ‘+’, ‘-’, ‘*’ y ‘/’.

2. En el siguiente árbol 2-3-4 borrar los elementos: 30, 15 y 20. (Primero se borra el 30, sobre el resultado obtenido se borra el 15 y, finalmente, sobre el resultado obtenido se borra el 20).

Criterios: Si hay que borrar un nodo con 2 hijos hay que sustituir por el menor de la derecha. Se tomará como *r* el hermano de la derecha.

Nota: Realizar el borrado de los elementos paso a paso indicando las transformaciones realizadas.



3. En un árbol izquierdista mínimo (leftist mínimo) inicialmente vacío, insertar los siguientes items: 7, 15, 11, 1, 2, 9, 3, 13, 6, 12, 4, 22, 10, 14, 5, 30, 21, 20, 23.

Del leftist minino anterior, realizar dos operaciones de borrado.

4. Calcula la complejidad temporal en el caso peor de las siguientes operaciones utilizando las estructuras que se indican.

Utiliza la Notación asintótica O (big-omicron).

El número de elementos le llamamos *n*.

No se permiten elementos repetidos.

Razona brevemente tu respuesta.

- Inserción y borrado en una lista enlazada no ordenada.
- Inserción y borrado en un vector ordenado circular.
- Inserción de una palabra en un trie con nodos terminales.
- Obtener mínimo de un heap mínimo.
- Insertar en un árbol binario de búsqueda enlazado.
- Borrar en un árbol leftist.

Examen TAD/PED junio 2005. Soluciones

1)

Sintaxis:

inversa: vector \rightarrow vector

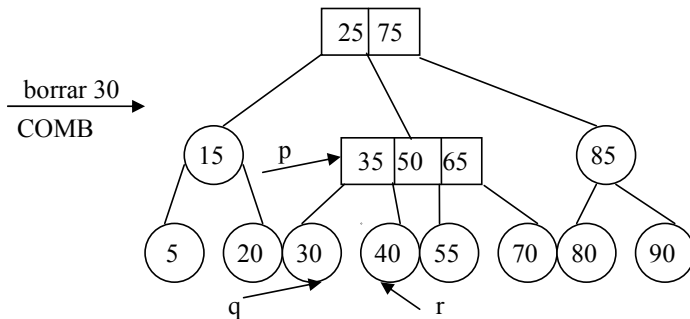
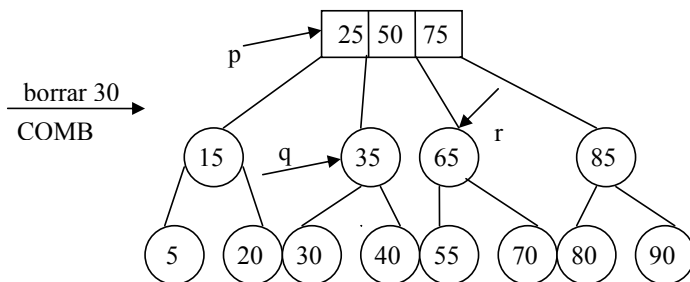
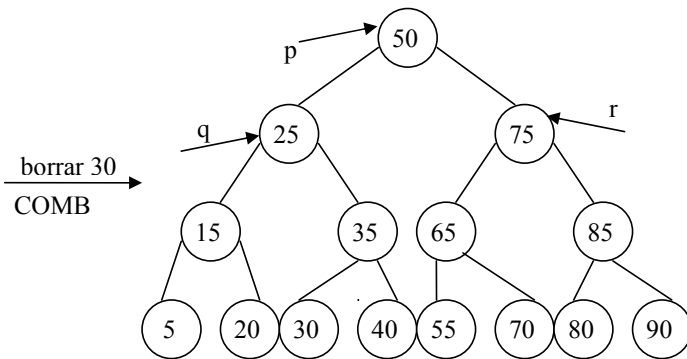
Semántica:

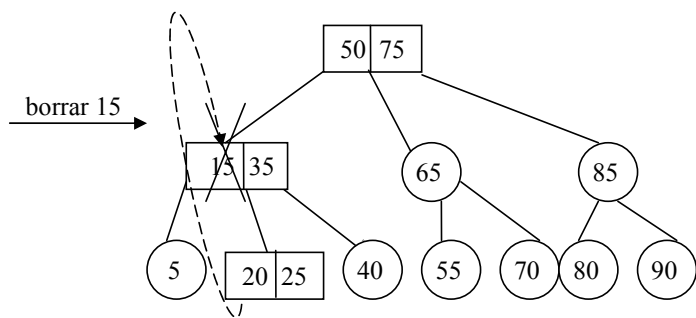
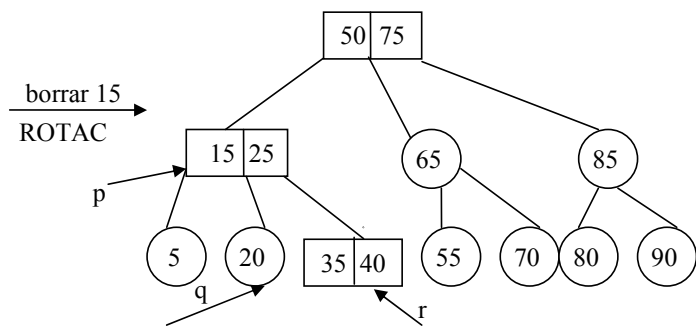
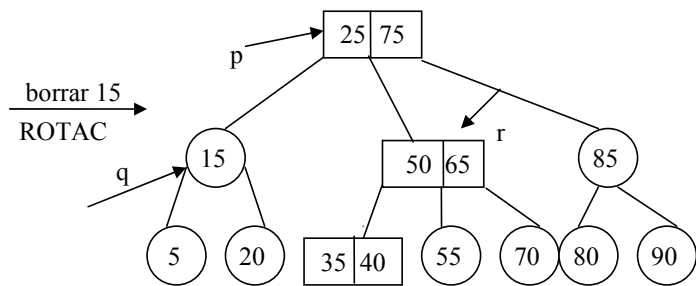
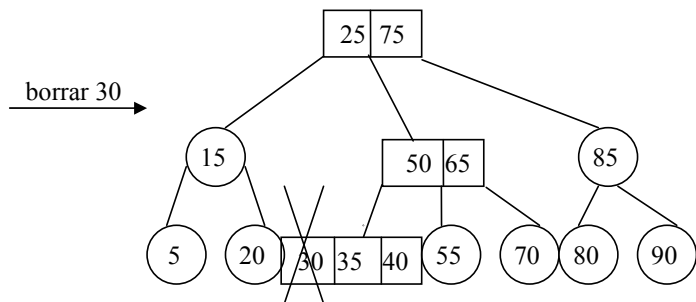
Var v: vector; i,x: natural;

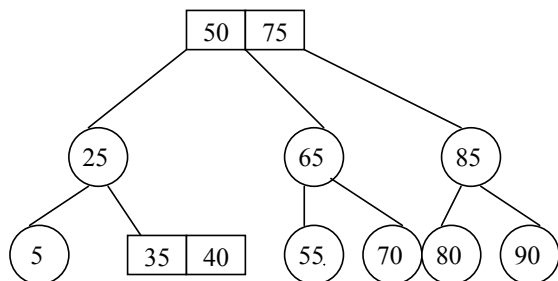
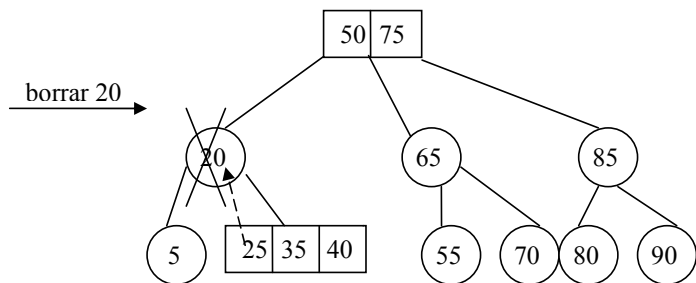
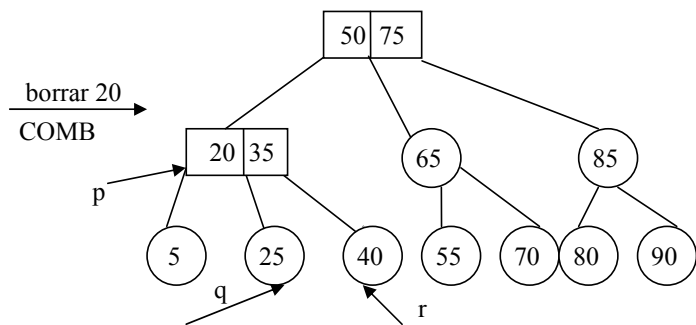
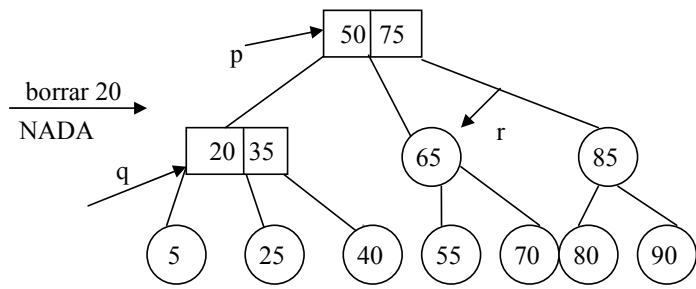
inversa(crear_vector()) = crear_vector() //caso base

inversa(asig(v,i,x)) = asig(inversa(v),100-i+1,x) //caso general

2)

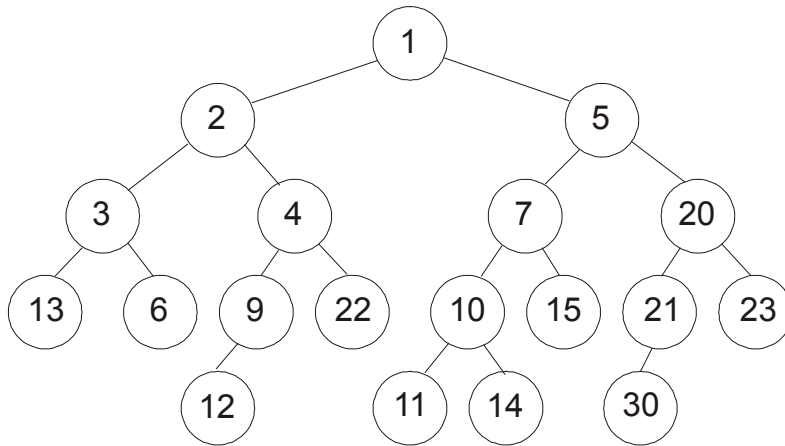




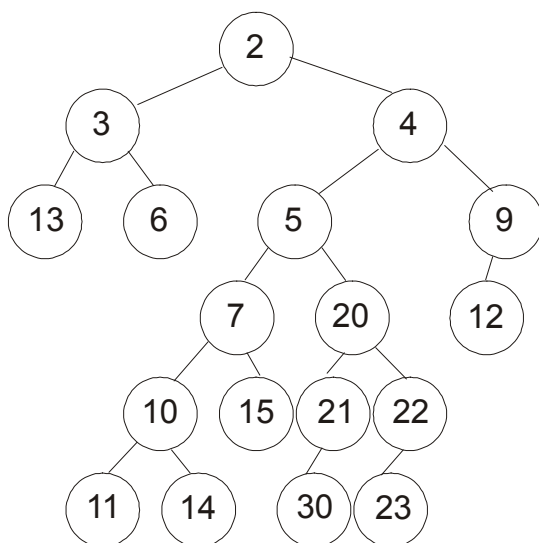


3) LEFTIST

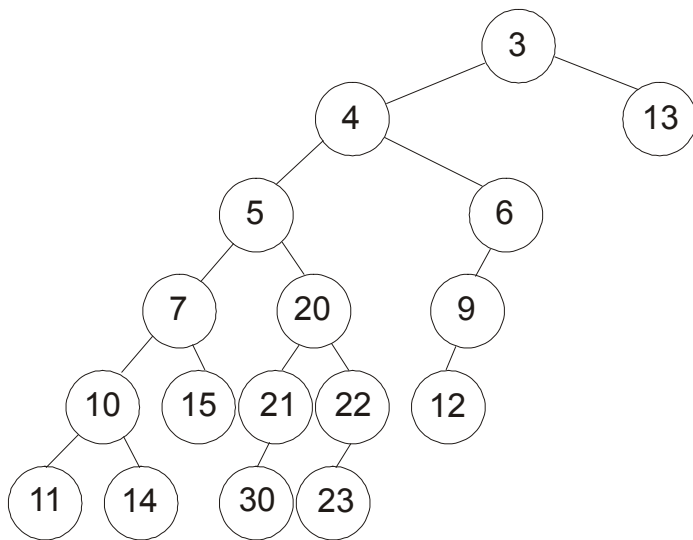
A)



B)



c)



4) COMPLEJIDAD

- Inserción y borrado en una lista enlazada no ordenada.
Inserción $O(n)$.
Borrado $O(n)$.
- Inserción y borrado en un vector ordenado circular.
Inserción $O(n)$.
Borrado $O(n)$.
- Inserción de una palabra en un trie con nodos terminales.
 $O(L)$. Siendo L la longitud de la palabra.
- Obtener mínimo de un heap mínimo.
 $O(1)$
- Insertar en un árbol binario de búsqueda enlazado.
 $O(n)$
- Borrar en un árbol leftist.
 $O(\log n)$.