Apellidos:
Nombre:
Convocatoria:
DNI:

Examen PED junio 2013
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		
Longitud: LISTA -> NATURAL Si L es una lista, a es un ítem de la lista: a = Longitud (L) es un uso sintácticamente correcto de la operación Longitud.	❑	■	1	F
Sea el método Primera perteneciente a la clase Tlista que devuelve la primera posición de la lista que lo invoca: TPosicion Tlista::Primera( ) class Tlista { { TPosicion p; public: ... p.pos = lis; private: return p; } Tnodo *lis; } En el método Primera se invoca al constructor y destructor para el objeto TPosicion p.	■	❑	2	V
El algoritmo de intercambio directo o burbuja estudiado en clase (ordenación de los elementos de un vector) tiene una complejidad de $\Omega(n^2)$, siendo *n* el número de elementos del vector.	■	❑	3	V
La operación BorrarItem, que borra todas las ocurrencias del item 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)	■	❑	4	V
Existe al menos un árbol binario, que representa los siguientes recorridos: inorden = YXZT, niveles = XTYZ.	❑	■	5	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.	■	❑	6	V
Un árbol completo siempre está balanceado respecto a la altura.	■	❑	7	V
El grado del árbol 2-3 es 2.	❑	■	8	F
En un árbol 2-3-4 sólo los nodos hoja y la raíz pueden ser de tipo 2-nodo.	❑	■	9	F
En los conjuntos representados como listas no ordenadas, la complejidad temporal de la operación “diferencia de conjuntos” es O(n), siendo *n* el número de elementos de cada conjunto.	❑	■	10	F
En la dispersión cerrada puede haber colisiones entre claves sinónimas y no sinónimas.	■	❑	11	V
La definición de un Heap Mínimo indica que ha de ser un árbol binario que además es árbol mínimo.	■	❑	12	F
En un grafo dirigido pueden existir infinitas aristas para un número “n” de vértices.	❑	■	13	F
Sea G un grafo no dirigido de n vértices. Si G tiene “n-1” aristas, entonces nunca podría tener un ciclo.	❑	■	14	F

Examen PED junio 2013

Normas:
- Tiempo para efectuar el ejercicio: **2 horas**
- En la cabecera de cada hoja **Y EN ESTE ORDEN** hay que poner: ***APELLIDOS, NOMBRE***.
- Cada pregunta se escribirá en hojas diferentes.
- 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.**
- **Las fechas de "Publicación de notas" y "Revisión del examen teórico" se publicarán en el Campus Virtual.**

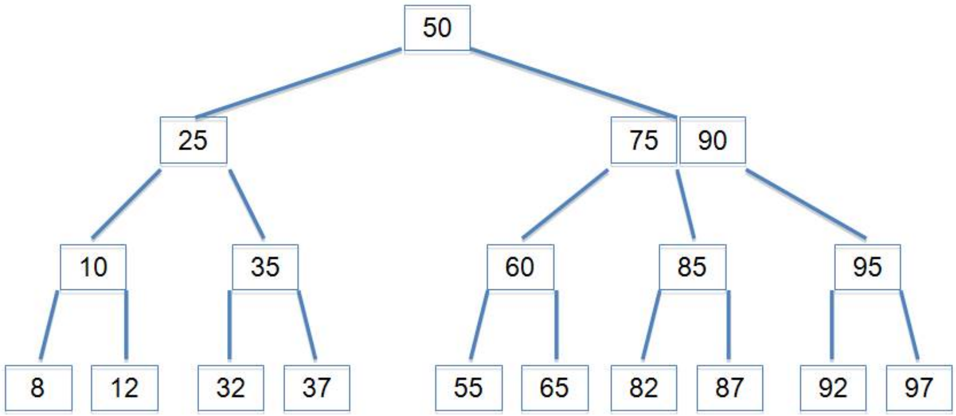
1.

a) Define la sintaxis y la semántica de la operación **inverso** que actúa sobre un vector de números naturales y devuelve el vector inverso del vector de entrada.

Nota: utilizar exclusivamente las operaciones constructoras generadoras del vector. Se asume que el tamaño del vector es una constante determinada, tam, y que están definidas y se pueden usar todas las operaciones de números naturales (suc, suma, resta, multiplicación y división).

b) Define la sintaxis y la semántica de la operación **insertar** vista en clase que inserta un elemento en una lista con acceso por posición.

2. Sea el siguiente árbol A:



a) Realiza el borrado de los ítems 8 y 35 del árbol A, suponiendo que este árbol es un árbol 234.
Criterio 1: r es el hermano de la izquierda.
Criterio 2: Si el ítem a borrar no está en una hoja, sustituir por el mayor de la izquierda.

b) Realiza el borrado de los ítems 8 y 35 del árbol A, suponiendo que este árbol es un árbol 23.
Criterio 1: r es el hermano de la izquierda.
Criterio 2: Si el ítem a borrar no está en una hoja, sustituir por el mayor de la izquierda.

3. Sea el siguiente montículo doble (DEAP) representado como un vector.

a) Inserta **sobre dicho vector** los siguientes elementos, 60, 1, 70, **mostrando y explicando** las operaciones realizadas **sobre el vector**.

b) Borra el máximo, mínimo y máximo sucesivamente del **<u>sobre el vector inicial</u> (sin insertar los elementos)**, **mostrando y explicando** las operaciones realizadas **sobre el vector**.

c) Indicar justificadamente el caso peor de la complejidad temporal de la inserción y del borrado en un DEAP **representado como un vector**.

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
-	*2*	*55*	*10*	*8*	*45*	*40*	*12*	*19*	*20*	*9*	*31*	*35*	*36*	*30*	*15*	*16*	*25*

Examen PED junio 2013. Soluciones

1.

a)

inverso(vector) → vector

VAR v: vector; i, x: natural;

inverso(crear_vector()) = crear_vector()

inverso(asig (v,i,x)) = asig(inverso(v), tam+suc(cero)-i, x)

b)

insertar(lista, posicion, item) → lista

VAR L_1: lista; x, y: item; p: posicion;

insertar(crear_lista( ), p, x) = crear_lista( )

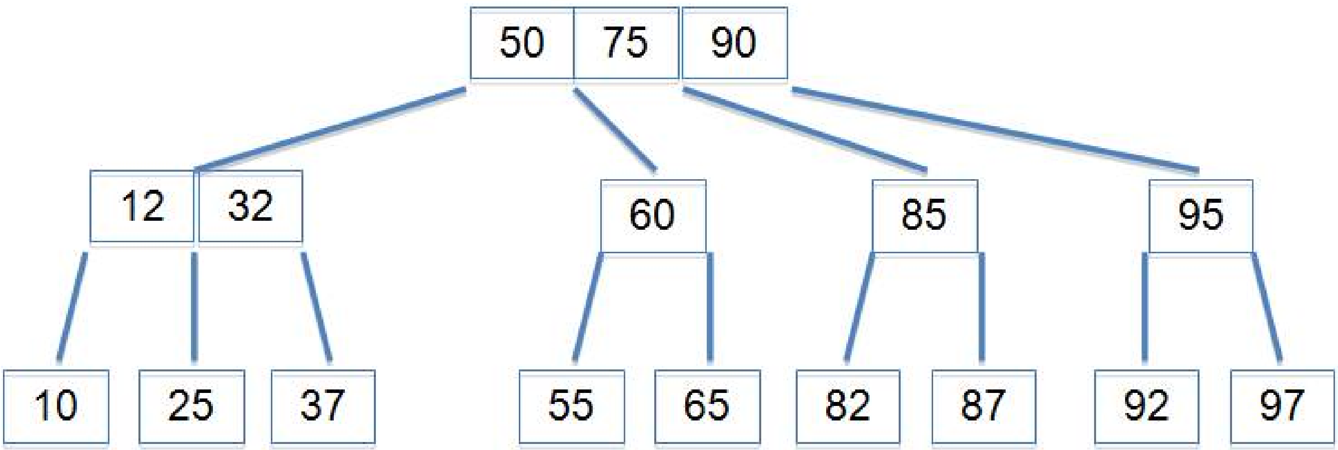
si p == primera(inscabeza(L_1, x)) **entonces**

insertar(inscabeza(L_1, x), p, y) = inscabeza(inscabeza(L_1, y), x)

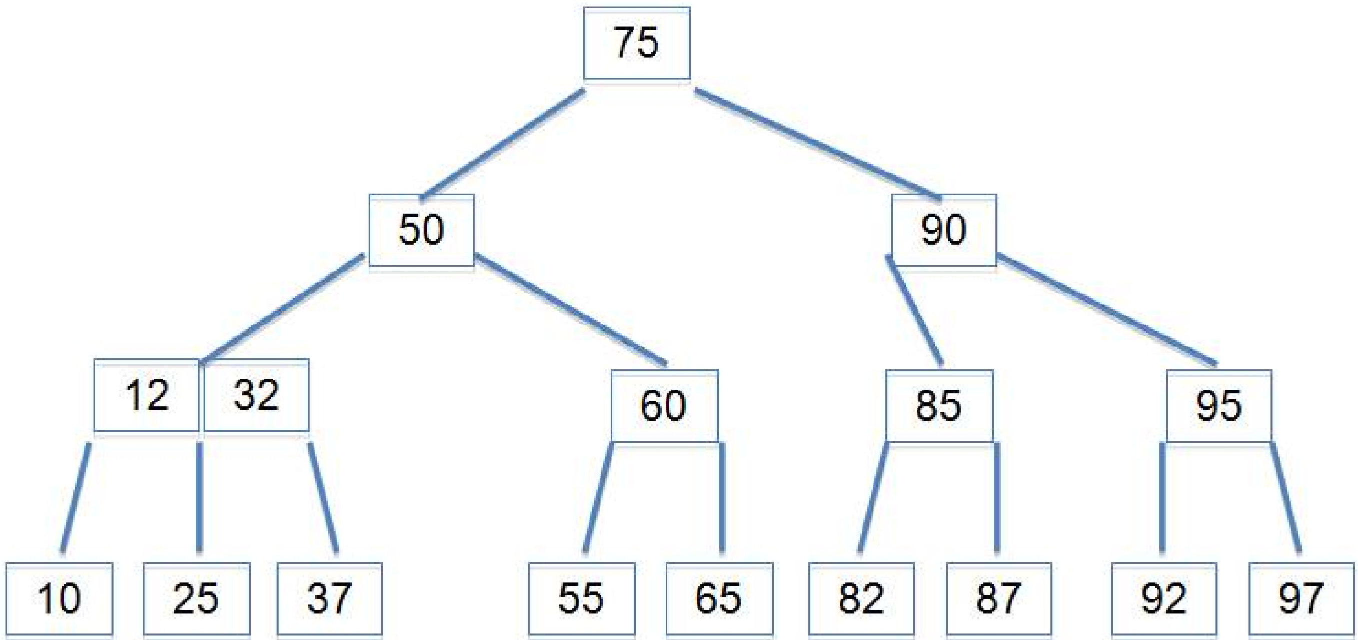
si no insertar(inscabeza(L_1, x), p, y) = inscabeza(insertar(L_1, p, y), x)

2.

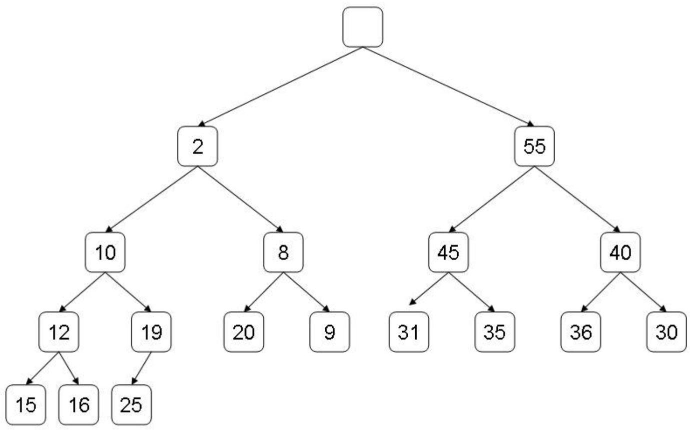
A)



B)



3.
a) El DEAP inicial sería:



1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*
-	2	55	10	8	45	40	12	19	20	9	31	35	36	30	15	16	25

Los movimientos de restructuración del DEAP se realizarán a partir del índice *i* de la casilla del vector en la que se encuentre el elemento a mover, con lo que se accederá a su padre accediendo a la casilla *i DIV 2*; a su hijo izquierdo accediendo a *i*2*; y a su hijo derecho accediendo a *i*2 + 1*.

Insertar 60

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
-	2	60	10	8	55	40	12	19	20	9	31	45	36	30	15	16	25	35

Insertar 1

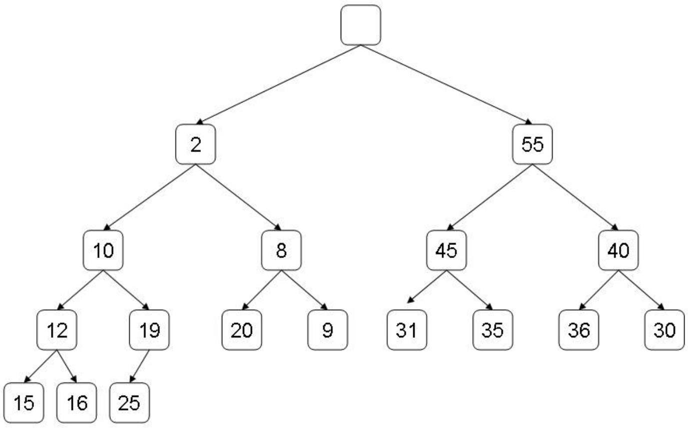
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-	1	60	10	2	55	40	12	19	8	9	31	45	36	30	15	16	25	35	20

Insertar 70

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
-	1	70	10	2	55	60	12	19	8	9	31	45	40	30	15	16	25	35	20	36

b)

El DEAP inicial sería:



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
-	2	55	10	8	45	40	12	19	20	9	31	35	36	30	15	16	25

Borrar máximo

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
-	2	45	10	8	35	40	12	19	20	9	31	25	36	30	15	16

Borrar mínimo

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-	8	45	10	9	35	40	12	19	20	16	31	25	36	30	15

Borrar máximo

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-	8	40	10	9	35	36	12	19	15	16	31	25	20	30

c) La complejidad de la inserción en un montículo doble en el peor caso es $O(\log_2 n)$, con "n" el número de elementos del DEAP. Como la cantidad de comparaciones que tenemos que hacer en el peor caso es igual al número de niveles que tiene el árbol, la complejidad en el peor caso estará en función de la altura del árbol. Como por definición un DEAP es un árbol binario completo, podemos afirmar que la altura de un DEAP es $O(\log_2 n)$.