Apellidos:
Nombre:
Convocatoria:
DNI:

Examen PED julio 2014. GRADO
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**		
La siguiente especificación corresponde a la operación de borrar una arista en un multigrafo: VAR G: Grafo; a, b, z, t: Vértice; p: Ítem; BorrarArista(CrearGrafo(), z, t) = CrearGrafo() BorrarArista(InsertarArista(G,a,b,p), z, t) = si (a == z) y (b == t) entonces BorrarArista(G, z, t) sino InsertarArista(BorrarArista(G, z, t), a, b, p)	■	❑	1	V
En la escala de complejidades, la complejidad logarítmica es menor que la lineal.	■	❑	2	V
El factor de carga de la dispersión abierta siempre está entre 0 y 1.	❑	■	3	F
En C++, los miembros protected son privados para el exterior, pero permiten el acceso a las clases derivadas.	■	❑	4	V
Para el siguiente algoritmo, la complejidad sería $O(n^2)$: *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];*	❑	■	5	F
La cota promedio de complejidad es el resultado de hacer la media entre la cota superior y la cota inferior.	❑	■	6	F
La operación de lista: Longitud: (LISTA) → NATURAL es una operación consultora.	■	❑	7	V
En la lista de acceso por posición (vista en clase) se establece un orden secuencial estricto a partir de las posiciones que ocupan sus elementos.	■	❑	8	V
La complejidad temporal del recorrido por niveles en un árbol binario es la misma que las de los recorridos in-pre-post orden.	■	❑	9	V
El mayor elemento en un árbol binario de búsqueda siempre se encuentra en un nodo hoja.	❑	■	10	F
Un árbol AVL es un árbol binario de búsqueda en el que la diferencia de nodos entre el subárbol izquierdo y derecho es como máximo uno.	❑	■	11	F
El mínimo número de elementos que se puede almacenar en un árbol 2-3 de altura h coincide con el número de elementos que hay en un árbol binario lleno de altura h.	■	❑	12	V
Existe un único árbol 2-3 de altura 3 que representa a las etiquetas del 1 al 9.	❑	■	13	F
Un árbol 2-3-4 es un árbol binario completo.	❑	■	14	F
La siguiente especificación algebraica corresponde a la operación unión de conjuntos: VAR A, B: Conjuntos; x:item Union(crear(), A) = A Union(Insertar(A,x),B)= si (Pertenece(B,x)) entonces Union(A, B) sino Insertar(Union(A, B), x)	❑	■	15	V
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.	■	❑	16	V
En un montículo doble todas las claves del montículo máximo son mayores que las del montículo mínimo.	❑	■	17	F
Al representar un grafo dirigido de N vértices y K aristas con una matriz de adyacencia, la matriz será simétrica respecto la diagonal principal.	❑	■	18	F
Los arcos de retroceso de un recorrido en profundidad de un grafo dirigido, nos indican la presencia de un ciclo.	■	❑	19	V

Examen PED julio 2014. GRADO

Normas:
- Tiempo para efectuar el examen: **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)** Utilizando exclusivamente las operaciones constructoras generadoras de vector, definir la sintaxis y la semántica de la operación *subvector* que actúa sobre un vector de números naturales y recibe dos números naturales (que representan posiciones del vector). La operación devuelve el vector original conteniendo sólo los elementos entre las posiciones dadas (ambas posiciones incluidas).
Nota: Se asume que las posiciones son correctas y están en el rango del vector: desde 1 hasta la longitud del vector. Se pueden utilizar todas las operaciones definidas para números naturales y booleanos.
Ej: *subvector* ([1,5,2,20,3], 2, 4) → [5,2,20]

b) Rellenar la siguiente tabla de complejidades temporales en su caso peor del Tipo Conjunto según se utilicen distintas representaciones para almacenar los elementos del mismo:

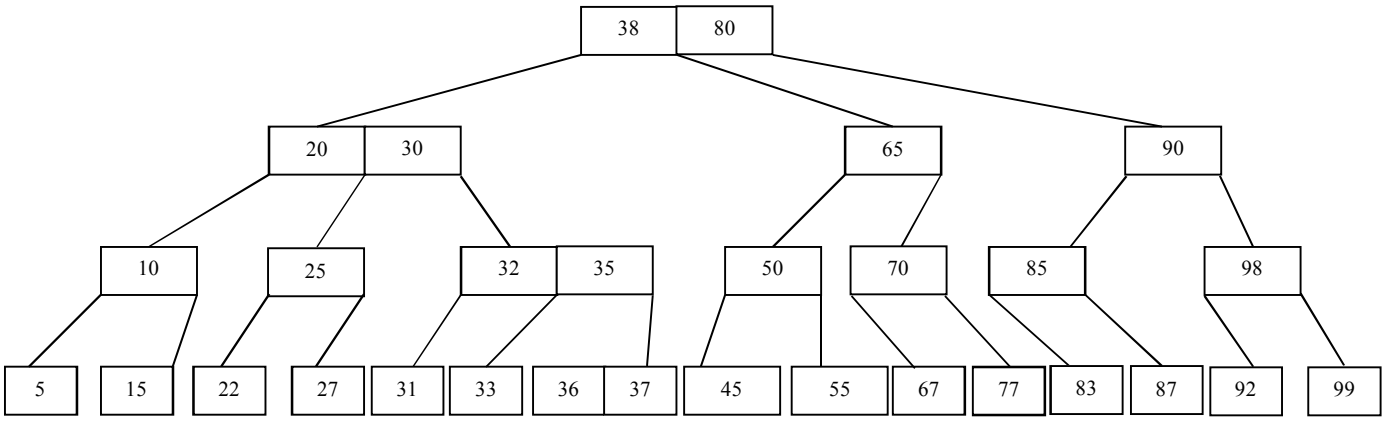
	ABB	**AVL**	**Árbol 2-3-4**	**Hash cerrado**
INSERCIÓN				
BÚSQUEDA				

Para que se valore la pregunta hay que justificar la respuesta para cada caso.
Nota: Las transformaciones realizadas en los árboles para la inserción de un elemento se asumen que tiene coste constante y, por lo tanto, no influyen en el cálculo de la complejidad.

c) Insertar en la siguiente tabla de dispersión cerrada de tamaño B=11, los siguientes elementos: 30, 3 y 16 utilizando una estrategia de redispersión de "segunda función hash".
Nota: hay que utilizar las fórmulas matemáticas para la inserción de elementos.

	12				27			19	20	
0	1	2	3	4	5	6	7	8	9	10

2. Dado el siguiente árbol A:



a) Realiza el borrado del ítem **80** del árbol A, suponiendo que este árbol es un **árbol 23**, indicando número y tipo de transformaciones utilizadas.
Criterio 1: Consultar 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) Del árbol original A, realizar un recorrido por niveles, y según ese recorrido, realizar **las quince primeras inserciones** de los elementos **en un DEAP** inicialmente vacío. Habrá que mostrar la secuencia de transformaciones realizadas, aunque se podrán realizar varias inserciones en la misma iteración.
c) Dada un nodo *i* (considerando la posición de la raíz como la posición 1, su hijo izquierda tendrá la posición 2, su hijo derecha la posición 3, y así sucesivamente) que se encuentra en el nivel *h* (la raíz tendrá nivel 1, sus hijos nivel 2 y así sucesivamente) de un DEAP de *n* elementos, calcular **razonadamente** la posición *j* simétrica a i (NO SE VALORARÁ SI SOLO SE PONE LA FÓRMULA)

Examen PED julio 2014. Grado. Soluciones

1.

a)

```
subvector: vector, natural, natural → vector
Var v: vector; i,x,p1,p2: natural
subvector(crear(), p1, p2) = crear()
subvector(asig(v,i,x), p1, p2) =
        si [(i>p1) y (i<p2)] o [i==p1] o [i==p2]
 entonces asig (subvector(v, p1, p2), i, x)
                sino subvector(v, p1, p2)
```

b)

	ABB	**AVL**	**Árbol 2-3-4**	**Hash cerrado**
INSERCIÓN	O(n)	$O(\log_2 n)$	$O(\log_2 n)$	O(1)
BÚSQUEDA	O(n)	$O(\log_2 n)$	$O(\log_2 n)$	O(1)

La inserción y la búsqueda tienen la misma complejidad ya que hay que buscar previamente el elemento a insertar en el conjunto (por definición, un conjunto no admite elementos repetidos).

ABB → lineal considerando el caso de un ABB degenerado (realmente sería una lista). En el peor de los casos, habría que recorrer la lista entera para insertar un elemento (comprobando que no hayan repetidos) o buscarlo. Complejidad lineal.

AVL y 2-3-4 → Tanto para las inserciones como las búsquedas, el peor caso sería recorrer toda la altura del árbol; ya que la altura es $h=\log_2(n+1)$, podemos concluir que la complejidad sería logarítmica.

Hash cerrado → Utilizaremos la información del elemento a almacenar para buscar su posición dentro de la estructura. Se calcula la función de dispersión y/o redispersión. Posteriormente, se realiza un acceso directo ya que es un vector. Estudios empíricos demuestran la complejidad constante.

c)

H(30)=30 MOD 11=**8**

k(30)=(30 MOD 10)+1=1

h1(30)=(8+1x1) MOD 11=**9**

h2(30)=(8+1x2) MOD 11=**10**

H(3)= 3 MOD 11= **3**

H(16)=16 MOD 11=**5**

k(16)=(16 MOD 10)+1=7

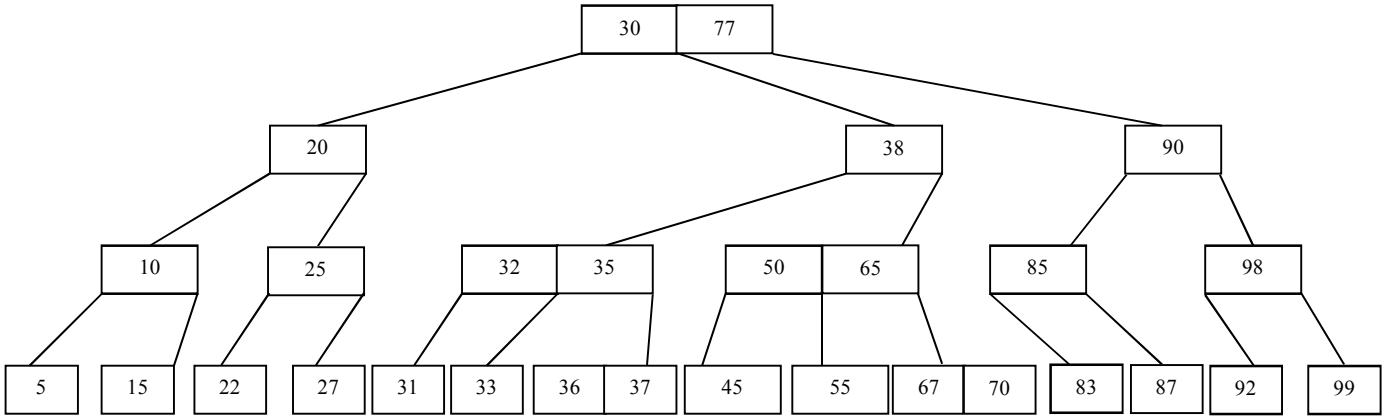
h1(16)=(5+7x1) MOD 11=**1**

h2(16)=(5+7x2) MOD 11=**8**

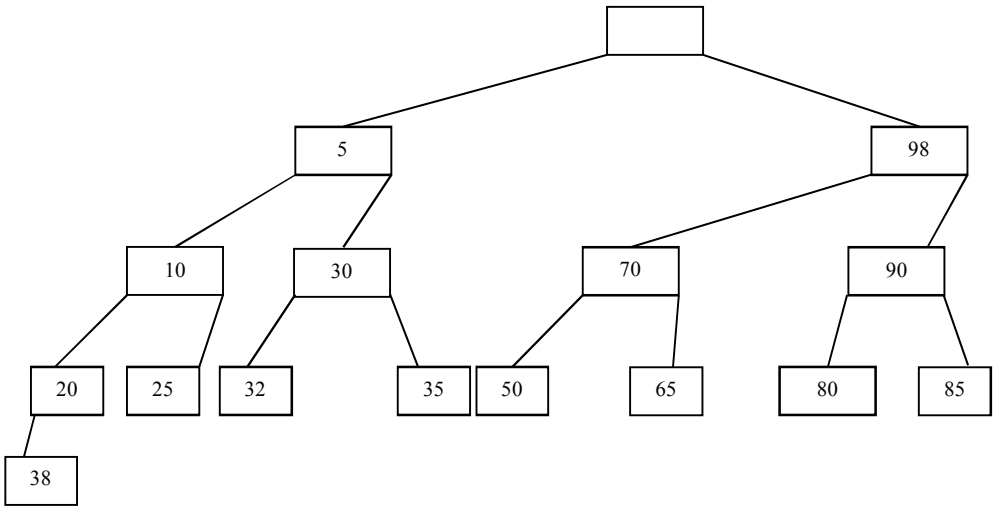
h3(16)=(5+7x3) MOD 11=**4**

2.

a) 2 combinaciones y 1 rotación



b)



c)

El número de nodos de un nivel *h* del DEAP será 2^{h-1}, con lo que dada una posición *i*, para calcular el simétrico, habrá que sumarle/restarle la mitad de esos nodos, es decir, $2^{h-1}/2 = 2^{h-2}$.

El nodo *i* estará en el montículo mínimo si $(i - (2^{h-1}-1) \leq 2^{h-2})$, considerando que el número de nodos del subárbol de altura *h-1* es $2^{h-1}-1$, por lo que $i - (2^{h-1}-1)$ representa la posición del nodo dentro del nivel *h*. Si esa posición dentro del nivel *h* es menor o igual que la mitad de nodos de ese nivel, entonces el nodo *i* estará en el montículo mínimo. Con ello, su simétrico estará en la posición $j=(i + 2^{h-2})$, pero habrá que controlar que exista ese simétrico: si $j > n$ entonces su simétrico será $j = j$ *DIV* 2, es decir el padre del simétrico estricto (el cual no existe en el DEAP).

Sino, el nodo *i* estará en el montículo máximo, con lo que su simétrico estará en la posición $j=i - 2^{h-2}$.