

Apellidos:

Nombre:

Convocatoria:

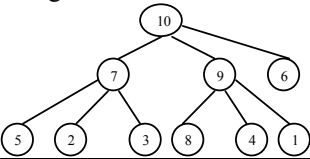
DNI:

## Examen PED junio 2015

### 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.
- Este test vale 2 puntos (sobre 10).
- **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 tratamiento de errores en la especificación algebraica, se añaden funciones constantes que devuelven un valor del tipo que causa el error.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
La complejidad temporal (en su caso peor) del siguiente fragmento de código es $O(n^2)$ <pre>int i, j, n, sum; for (i = 4; i &lt; n; i++) {     for (j = i-3, sum = a[i-4]; j &lt;= i; j++) sum += a[j];     cout &lt;&lt; "La suma del subarray " &lt;&lt; i-4 &lt;&lt; " es " &lt;&lt; sum &lt;&lt; endl; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	2	F
Es posible obtener una representación enlazada de una cola utilizando un único puntero que apuntará al fondo de la cola.	<input type="checkbox"/>	<input type="checkbox"/>	3	V
Dado un único recorrido de cualquier árbol, siempre es posible reconstruir dicho árbol	<input type="checkbox"/>	<input type="checkbox"/>	4	F
El coste temporal en su peor caso de insertar una etiqueta en un árbol binario de búsqueda es logarítmica respecto a la altura del árbol	<input type="checkbox"/>	<input type="checkbox"/>	5	F
La complejidad temporal en el peor caso y en el mejor caso de la operación inserción en un AVL son lineal y logarítmica respecto al número de nodos en el árbol	<input type="checkbox"/>	<input type="checkbox"/>	6	F
El borrado en un árbol AVL puede requerir una rotación en todos los nodos del camino de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	7	V
Dado un árbol 2-3 de altura h con n items: $2^h - 1 \leq n \leq 3^h - 1$	<input type="checkbox"/>	<input type="checkbox"/>	8	V
Los nodos hoja de un árbol 2-3 han de estar en el mismo nivel del árbol	<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 2-3-4 es un árbol 4-camino de búsqueda	<input type="checkbox"/>	<input type="checkbox"/>	11	V
La especificación algebraica de la siguiente operación indica que se devolverá el número de elementos del conjunto multiplicado por 3 (Operación(Conjunto) $\rightarrow$ Natural; Var: C: Conjunto; x: Ítem): Operación(Crear) $\Leftrightarrow 1$ Operación (Insertar(C, x)) $\Leftrightarrow 3 + \text{Operación}(C)$	<input type="checkbox"/>	<input type="checkbox"/>	12	F
En un montículo el número de claves en el hijo izquierda de la raíz es mayor o igual que en su hijo derecha	<input type="checkbox"/>	<input type="checkbox"/>	13	V
El siguiente árbol es un montículo máximo: 	<input type="checkbox"/>	<input type="checkbox"/>	14	F
La siguiente secuencia de nodos de un grafo es un ciclo: 1,2,3,2,1	<input type="checkbox"/>	<input type="checkbox"/>	15	F
Un bosque extendido en profundidad de un grafo dirigido al que se le añaden los arcos de retroceso es un grafo acíclico dirigido.	<input type="checkbox"/>	<input type="checkbox"/>	16	F

## Examen PED junio 2015

- 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
  - **Cada pregunta vale 2 puntos (sobre 10).**
  - Las fechas de “Publicación de notas” y “Revisión del examen teórico” se publicarán en el Campus Virtual.

**1.** Dado el grafo **no dirigido** representado por la lista de adyacencia que se muestra a continuación (se recorrerá cada lista de adyacencia considerándola ordenada de menor a mayor):

**a)** Obtener DFS(1), el árbol extendido en profundidad partiendo del vértice 1 y la clasificación de las aristas.

**b)** Obtener BFS(1).

1 → 2 → 4

2 → 9

3 → 1 → 7

5 → 6 → 3 → 1

6 → 1

7 → 1 → 5

8 → 7

9 → 1

10 → 14

11 → 10

12 → 11 → 13 → 10

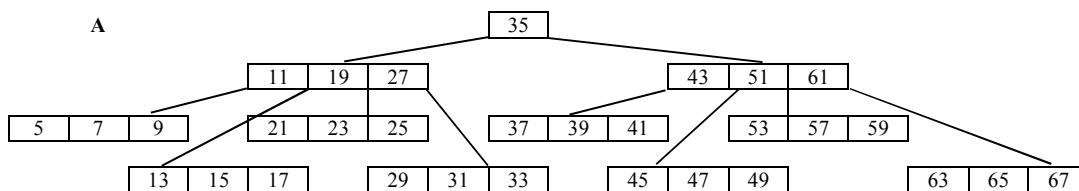
13 → 10 → 11

14 → 11 → 12 → 13

**c)** Utilizando exclusivamente las operaciones constructoras generadoras de grafo, definir la sintaxis y la semántica de la operación *CalculaPesos* que actúa sobre un grafo dirigido ponderado donde: los vértices son números Naturales y los Item son números Naturales que representan distancias kilométricas para cada par de vértices. *CalculaPesos* devuelve la suma de las distancias kilométricas de todos los arcos del grafo entre vértices pares.

Nota: se pueden utilizar todas las operaciones definidas para números naturales.

**2.** Sea el siguiente árbol 2-3-4:



**a)** Sobre este árbol A, realiza la inserción de los ítems 1, 24 y 58 en ese orden.

**b)** Escribe un árbol 2-3-4 de altura 4 en el que al borrar el ItemIzquierda del nodo raíz, haya que realizar 2 combinaciones y 1 rotación (da igual el orden en que se realicen) considerando los siguientes criterios. Realizar dicho borrado.

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.** Dada la siguiente función que calcula el código ASCII de una cadena de caracteres:

```
int hashChar(char *cadena, int max=20000){
    int cod=0;
    for (int i=0; i<strlen(cadena); i++)
        if (cod<max)
            cod=cod+toascii(cadena[i]);
        else
        {
            cod=cod%100;
            cod=cod+toascii(cadena[i]);
        }
    return(cod);
}
```

Códigos ASCII de cada letra:

A=65	C=67	E=69	J=74	M=77
N=78	O=79	P=80	R=82	S=83

**a)** Indicar la complejidad temporal. ¿Existe mejor y peor caso?

**b)** Si no se utilizara el parámetro max, es decir, que dentro del for sólo estuviera la instrucción *cod = cod + toascii(cadena[i]);* ¿Habría algún caso en que la función hashChar no funcionara correctamente?

**c)** Insertar en una tabla de dispersión cerrada de tamaño B=7, con función de dispersión  $H(x) = x \text{ MOD } B$  y con estrategia de redispersión segunda función hash, la siguiente secuencia de elementos:

- 1) AMOR(303) – 2) COSA(294) – 3) MORA(303)
- 4) ESPONJA(528) – 5) RAMO(303) –
- 6) JAPONES(528) – 7) ROMA(303) –
- 8) CASO(294)

¿Qué elemento o elementos no se podrían insertar en la tabla? ¿Por qué?

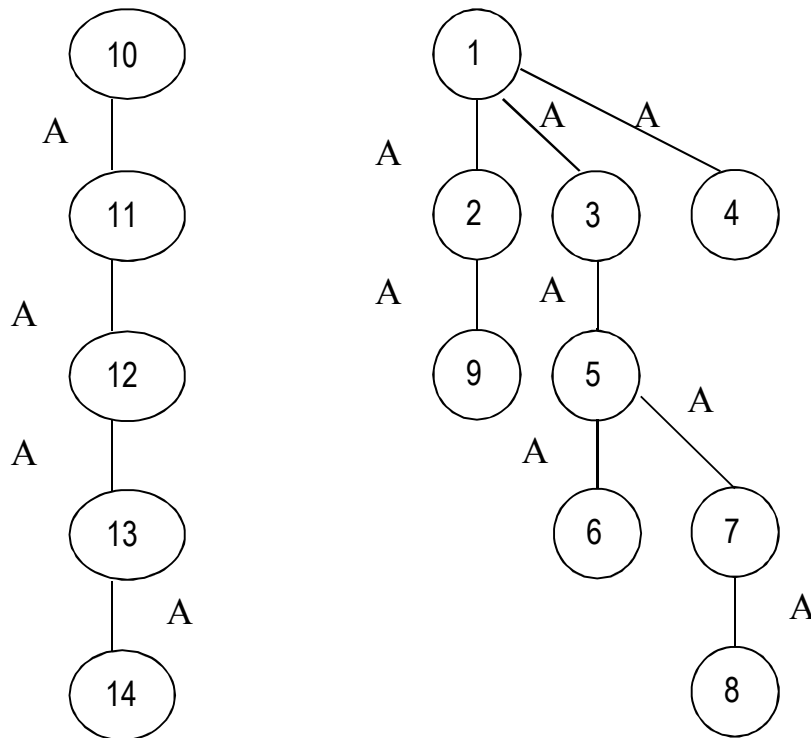
Justifica tus respuestas.

## Examen PED junio 2015. Soluciones

1.

a) DFS(1)=1,2,9,3,5,6,7,8,4. Se continúa por DFS(10)=10,11,12,13,14

Árbol extendido en profundidad. Las aristas marcadas son de árbol (A), el resto son de retroceso



b) BFS(1)=1,2,3,4,5,6,7,9,8

c)

CalculaPesos: grafo  $\rightarrow$  natural

Var G: grafo; x,y: vértice; p: natural;

CalculaPesos (crear\_grafo())=0

CalculaPesos (InsertarArista(G,x,y,p))=

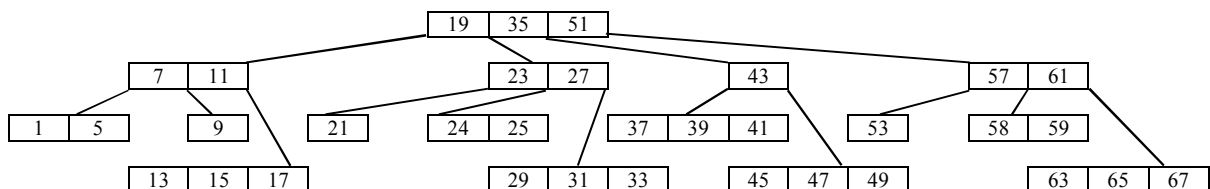
si ((x MOD 2 == 0) && (y MOD 2 == 0))

entonces p + CalculaPesos (G)

sino CalculaPesos (G)

2.

a)



b) Por ejemplo, el ejercicio 1 de los apuntes

3.

a)  $\Omega(n)=1+1*n=n$  (en el caso de que entrara en el if)

$O(n)=1+2*n=n$  (en el caso de que entrara en el else)

La diferencia entre ambos casos es de sólo un paso por lo que podemos considerar que el caso mejor y peor coinciden.

- b) Si se pasara como parámetro una cadena muy larga y la suma de los códigos ASCII de sus letras superara el tamaño máximo de almacenamiento del tipo int se produciría un error.
- c) Dado que es una tabla de dispersión cerrada de tamaño B=7 como máximo se podrían insertar 7 elementos, por lo que el último elemento CASO, no se podría insertar hasta que no se eliminase otro elemento de la tabla.

2ª función hash  $k(x) = (x \text{ MOD } (B-1)) + 1$   
 $h_i(x) = (h_{i-1}(x) + k(x)) \text{ MOD } B$

0	COSA
1	ROMA
2	AMOR
3	ESPONJA
4	RAMO
5	JAPONES
6	MORA

AMOR:  $H(303)=303 \text{ MOD } 7=2$

COSA:  $H(294)=294 \text{ MOD } 7=0$

MORA:  $H(303)=303 \text{ MOD } 7=2$

$k(303)=(303 \text{ MOD } 6)+1=4$

$h_1(303)=(2+4) \text{ MOD } 7=6$

ESPONJA:  $H(528)=528 \text{ MOD } 7=3$

RAMO:  $H(303)=303 \text{ MOD } 7=2$

$k(303)=(303 \text{ MOD } 6)+1=4$

$h_1(303)=(2+4) \text{ MOD } 7=6$

$h_2(303)=(6+4) \text{ MOD } 7=3$

$h_3(303)=(3+4) \text{ MOD } 7=0$

$h_4(303)=(0+4) \text{ MOD } 7=4$

JAPONES:  $H(528)=528 \text{ MOD } 7=3$

$k(528)=(528 \text{ MOD } 6)+1=1$

$h_1(528)=(3+1) \text{ MOD } 7=4$

$h_2(528)=(4+1) \text{ MOD } 7=5$

ROMA:  $H(303)=303 \text{ MOD } 7=2$

$k(303)=(303 \text{ MOD } 6)+1=4$

$h_1(303)=(2+4) \text{ MOD } 7=6$

$h_2(303)=(6+4) \text{ MOD } 7=3$

$h_3(303)=(3+4) \text{ MOD } 7=0$

$h_4(303)=(0+4) \text{ MOD } 7=4$

$h_5(303)=(4+4) \text{ MOD } 7=1$