

Apellidos:	Nombre:	Grupo:
Cognoms:	Nom:	Grup:

Estructuras de datos
1º Ingeniería informática
(Junio 1999)
Cuestiones

A

Tiempo: 1 hora

No se permiten libros ni apuntes.

La puntuación para preguntas con múltiples opciones es:

- Pregunta correcta: 1 punto
- Pregunta incorrecta: -0,2 puntos
- Pregunta en blanco: 0 puntos

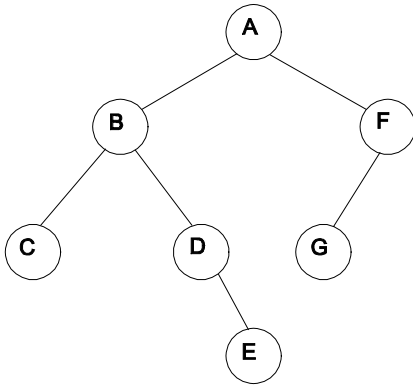
Las preguntas en las que se marquen varias opciones serán consideradas incorrectas.

El examen se responderá en la misma hoja en el espacio reservado para ello. cualquier respuesta fuera de estas hojas será ignorada.

- 1- ¿Cuál de las siguientes afirmaciones es cierta referida a un montículo de máximos?
- (a) El valor mínimo está en el nodo más a la derecha del árbol
 - (b) El valor mínimo está en el nodo más a la izquierda del árbol
 - (c) El valor mínimo está en un nodo terminal
 - (d) El valor mínimo está en el último nivel del árbol
- 2- ¿Cuál será el coste espacial (ocupación de memoria) de una cola de 40 elementos enteros representada mediante un array de 100 elementos y mediante una representación dinámica?

- 3- Un tipo abstracto de datos especifica:
- (a) La representación en memoria de la información
 - (b) La representación en memoria de la información y las operaciones asociadas
 - (c) Las operaciones válidas para un tipo de datos y sus propiedades
 - (d) La declaración de tipos en un lenguaje de programación
- 4- Se desea invertir una cadena de caracteres. ¿Qué estructura de datos auxiliar nos ayudaría a resolver el problema de manera más eficiente?
- (a) Una pila.
 - (b) Una cola.
 - (c) Un montículo de máximos donde la prioridad es el índice del carácter en la cadena.
 - (d) Un árbol binario de búsqueda organizado por índices del carácter en la cadena.
- 5.- Un árbol binario con 1023 nodos y altura 10 tiene:
- (a) 256 nodos terminales
 - (b) 512 nodos terminales
 - (c) 511 nodos terminales
 - (d) El número de nodos terminales es igual al número de nodos de grado uno más uno.
- 6 El recorrido de un árbol binario de búsqueda en orden infijo nos proporciona la secuencia ordenada de sus elementos, con lo que podemos utilizar el árbol como una estructura auxiliar para ordenar un array de 'n' enteros. ¿Cuál sería el coste de esta ordenación?
- (a) $O(n)$
 - (b) $O(n * (\lg n + n))$
 - (c) $O(n * \lg n + n)$
 - (d) $O(\lg n)$

7- Escribir la matriz de adyacencia que representa el grafo no dirigido y completo cuyo árbol de recorrido BFS es:



8- Una lista no vacía está formada por un primer elemento, llamado *head* (cabeza), y el resto de la lista, llamado *tail* (cola). Por ejemplo, si $L = (a_1, a_2, a_3, \dots, a_n)$ es una lista, entonces **Head** (L) es a_1 y **Tail** (L) es $(a_2, a_3, \dots, a_n)$. Sabiendo esto, indicad qué hace la siguiente función (siendo T el tipo de datos contenido en la lista)

```

Function Examen ( x: T; L: Lista ): Boolean;
Begin
  If Vacía ( L ) Then Examen := FALSE
  Else
    If Head ( L ) = x Then Examen := TRUE
    Else Examen := Examen ( x, Tail ( L ) )
End;
  
```

9- Supongamos un vector con 100 elementos enteros que está ordenado. Si a ese vector le añadimos al final tres nuevos elementos, tal que el vector queda parcialmente desordenado, ¿cuál de los siguientes métodos de ordenación sería más eficiente para volver a ordenar el vector?

- (a) Intercambio directo
- (b) Inserción directa
- (c) Selección directa
- (d) Método de Prim

10- Suponiendo la definición de las funciones **Head** y **Tail** vistas en el ejercicio 7, y suponiendo la existencia de una nueva función **Construir** (x, l) que construye una nueva lista cuya cabeza es x y cuya cola es l (inserta el elemento x al principio de la lista), di qué realiza la siguiente función, si las listas l_1 y l_2 están inicialmente ordenadas:

```

Function XX ( l1, l2: Lista ): Lista;
Begin
  If Vacía ( l1 ) Then XX := l2
  Else
    If Vacía ( l2 ) Then XX := l1
    Else
      If Head ( l1 ) < Head ( l2 ) Then
        XX := Construir ( Head ( l1 ), XX ( Tail ( l1 ), l2 ) )
      Else
        XX := Construir ( Head ( l2 ), XX ( l1, Tail ( l2 ) ) )
    End
  End
End
  
```

Estructuras de dades
1º Ingeniería Informática
(Junio 1999)
Problemas

A

Temps: 1 hora 30 minutos

No se permiten ni libros ni apuntes

Problema 1:

Realizar un procedimiento en Pascal que realice la operación unión de dos conjuntos:

Procedure union (var C, D: conjunto);

La operación **union** realiza la unión de C y D, guardando el resultado en C y dejando D igual al conjunto vacío:

union(C,D)= (C ← C ∪ D = { x / x ∈ C ∨ x ∈ D }) ∧ (D=∅)

Ejemplo:

Si C={1, 3, 5, 7} y D={1, 2, 3, 4, 5} entonces

union(C,D) → C={ 1, 2, 3, 4, 5, 7} y D={}

El tipo conjunto se supone implementado mediante listas doblemente enlazadas y ordenadas.

Problema 2:

Realizar en Pascal la operación **insertar** en un árbol binario de búsqueda, suponiendo una representación enlazada de los árboles donde en cada nodo se guarda un dato y la información necesaria para acceder a sus hijos y a su padre. Especificar claramente las declaraciones de tipo correspondiente. Se deben implementar todas las rutinas que se utilicen en este ejercicio.