



Departamento de Estadística, Matemáticas e Informática
Área de Lenguajes y Sistemas Informáticos

ESTRUCTURAS DE DATOS

2º Grado en Ingeniería Informática en Tecnologías de la Información

EXAMEN ORDINARIO CONVOCATORIA JUNIO

Duración de la prueba: **60 minutos**

Cada pregunta viene etiquetada con su puntuación máxima

Nombre Alumno: _____

DNI: _____ . Firma: _____

1.- (1) Sobre el proceso de abstracción en TADs

a) ¿En qué consiste el proceso de abstracción sobre funciones que permiten los TAD?

b) ¿Qué tipos de abstracciones permite un TAD?

2.- (0,5) El TAD tiene dos partes muy diferenciadas: Interfaz e Implementación. ¿Qué es la implementación?

Es el lugar en donde se aclara cómo realizar las operaciones del TAD, es por ello que sólo es visible al diseñador y al programador.

Fuente: PDF Tema 2 Abstracciones.

3.- (1,5) Sea la siguiente lista enlazada:

L.ini → 25 → 30 → 45 → 60 → 65 → 80 → 90 → NULL

↑
A

↑
B

Escribir

- a) L.sig.valor ?
- b) A.sig.valor ?
- c) Tras hacer A=B.sig
A.valor ?
A.sig ?



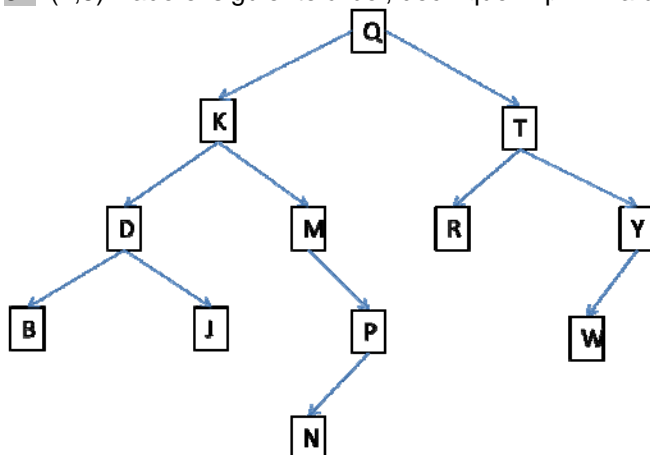
4.- (1) TAD Pila, especificación, Sintaxis

5.- (1) TAD Lista. ¿Qué dos tipos de implementación conoces respecto al tipo de memoria utilizada? Explica brevemente ambos tipos

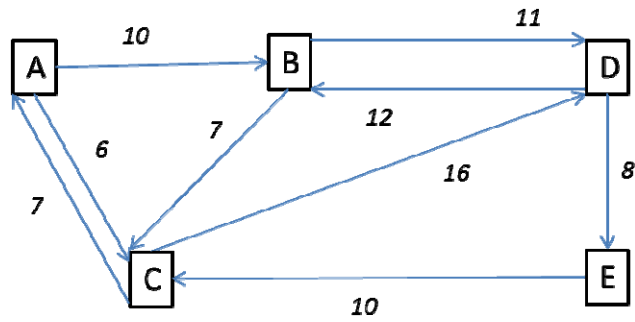
6.- (0,5) Cita 4 operaciones típicas entre conjuntos

7.- (0,5) ¿Para qué suelen emplearse los árboles TRIE?. Pon algún caso de ejemplo

8.- (1,5) Dado el siguiente árbol, decir qué imprimiría un recorrido inorden



9.- (1,5) Dado el siguiente grafo,



- ¿Se trata de un grafo dirigido? ¿y etiquetado? ¿y pesado?
- Dibuja la correspondiente matriz de adyacencia

10.- (1) Los algoritmos de Prim y de Kruskal, ¿qué tipo de problemas resuelven?

ESTRUCTURAS DE DATOS
2º Grado en Ingeniería Informática en Tecnologías de la Información

Septiembre, 2014
Duración de la prueba: **60 minutos**

Apellidos: _____ Nombre: _____

DNI: _____ Firma: _____

1. Sobre el proceso de abstracción en TADs
 - a) ¿En qué consiste el proceso de abstracción sobre funciones que permiten los TAD?
 - b) ¿Qué tipos de abstracciones permite un TAD?

2. ¿Qué dos tipos de especificaciones formales de un TAD existen?

3. Enumera las partes que tiene una especificación de un TAD

4. ¿En qué fase o fases de la especificación de un TAD se diferencia la especificación formal de la informal?

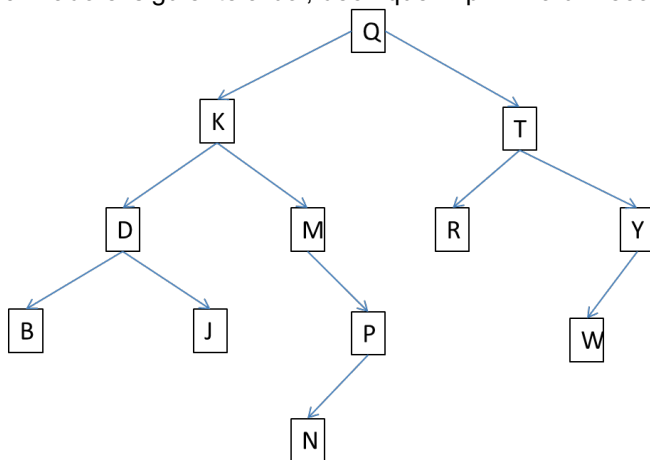
5. Enumera los tipos de lista que conoces

6. Especificación informal de TAD Cola. Semántica

7. Cita 4 operaciones típicas entre conjuntos

8. ¿Para qué suelen emplearse los árboles TRIE?. Pon algún caso de ejemplo

9. Dado el siguiente árbol, decir qué imprimiría un recorrido pre-orden



10. Enumera al menos 3 problemas clásicos a resolver mediante grafos

ESTRUCTURAS DE DATOS
2º Grado en Ingeniería Informática en Tecnologías de la Información
ELCHE

17, Junio, 2014

Duración de la prueba: **60 minutos**
Preguntas de la 1 a la 8 (1 punto)
Pregunta 9 (2 puntos)

Nombre: _____	
Apellidos: _____	
DNI: _____	FIRMA: _____

1. El TAD tiene dos partes muy diferenciadas: Interfaz e Implementación.
¿De qué se encarga cada una de ellas?

Interfaz

Consiste en una lista de operaciones junto con sus argumentos (es decir, el tipo de los parámetros de entrada y el tipo del resultado), y es la única parte visible al usuario del TAD.

Implementación

Es el lugar en donde se aclara cómo realizar las operaciones del TAD, es por ello que sólo es visible al diseñador y al programador.

2. Dí en qué condiciones es mejor implementar las listas con arrays y cuando es mejor usar memoria dinámica.

Arrays: Es más rápida y se usa cuando ya se conoce el tamaño de la lista y sabemos que no va a cambiar.

Memoria Dinámica: Cuando no conoces el tamaño de la lista que vas a utilizar.

3. ¿En qué se diferencian fundamentalmente el TAD Lista del TAD conjunto?

TADLista: Es una secuencia ordenada de elementos (según el orden que ocupan en la lista) de un tipo de datos (tipo base) dado.

TADConjunto: El orden no es importante, no hay orden.

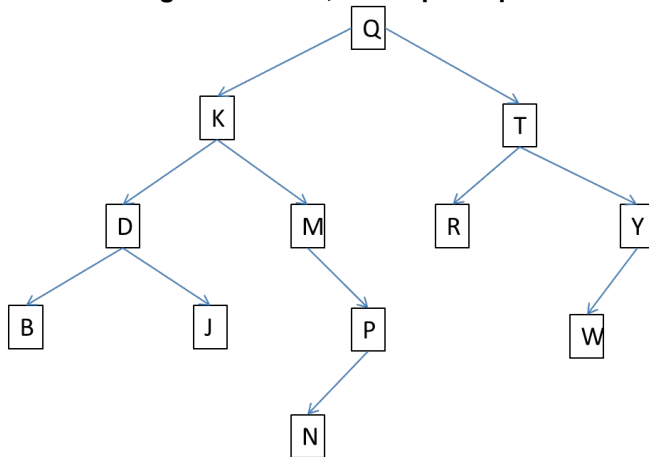
4. Define árbol binario de búsqueda

Un ABB es un tipo de árbol ordenado donde cada nodo tiene asociado un valor de clave y este valor de clave es mayor o igual que los valores de clave de los nodos de su subárbol izquierdo y menor o igual que todos los de su subárbol derecho.

5. ¿Cuál es el número mínimo de niveles que puede tener un árbol binario de búsqueda con 100 nodos?. Dibuja un esquema del mismo

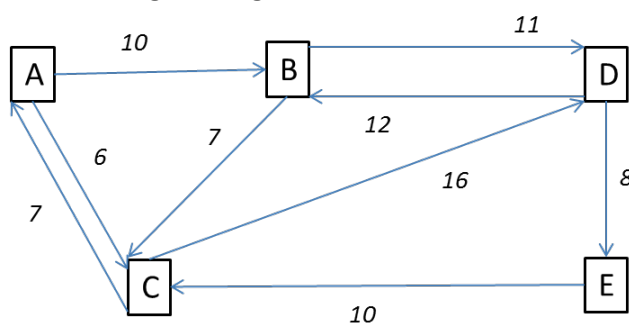
El nivel mínimo es de 7, ya que por cada nivel se va multiplicando * 2 el numero de nodos.

6. Dado el siguiente árbol, decir qué imprimiría un recorrido in-orden



B-D-J-K-N-P-M-Q-R-T-W-Y

7. Dado el siguiente grafo,



a) ¿Se trata de un grafo dirigido? Si ¿y etiquetado? Si¿y pesado?Si

b) Dibuja la correspondiente matriz de adyacencia

c) ¿Se trata de un grafo conexo (o conectado)? Si ¿Por qué? Por que hay un camino entre cualquier par de vértices.

d) ¿es fuertemente conexo? Si ¿Por qué? Por que es conexo y dirigido.

e) ¿es completo? No¿Por qué? Porque no existe una arista entre cualquier par de vértices.

8. Los algoritmos de Prim y de Kruskal, ¿qué tipo de problemas resuelven?
Los problemas de coste total mínimo.

9. Práctica 2: Escribe la especificación formal de la función `BuscarNodo` sobre un TAD ABB, teniendo en cuenta la siguiente interfaz (main)

```
main()
{
    ABB arbol = NULL;
    int x;
    char opcion;

    do
    {
        system("cls");
        cout << "\n\t\t\t\t\t \ARBOL BINARIO DE BUSQUEDA (ABB) ";
        cout << "\n\t\t\t\t\t-----";
        cout << "\n\n\t 1) Insertar nodo";
        cout << "\n\t 2) Buscar nodo";
        cout << "\n\t 3) Imprimir \xA0rbol";
        cout << "\n\t 4) Eliminar nodo";
        cout << "\n\t 5) N\xA3mero total de nodos";
        cout << "\n\t 6) Eliminar sub\xA0rbol";
        cout << "\n\n\t 0) Salir ";
        cout << "\n\n\t\t\t\t\t-----";

        opcion=getch();

        switch (opcion)

        ( . . . )
    }
}
```



```
case '2':
    if (arbol!=NULL)
    {
        system("cls");
        cout << "\n\t 2) Buscar nodo";

        if (BuscarNodo(arbol,x))
        {
            cout << "\n\n\t * El nodo SI esta";
            cout << "\n\n\n\t\t <pulse cualquier tecla ";
            getch();
        }

        else
        {
            cout << "\n\n\t * El nodo NO esta";
            cout << "\n\n\n\t\t <pulse cualquier tecla ";
            getch();
        }

    }
    else
    {
        cout << "\n\n\t * árbol vacio      }
    break;
( . . . )
```