



UNIVERSIDAD DE MÁLAGA
E.T.S.I. TELECOMUNICACIÓN

Programación modular
19 de junio de 2003
(1º de Ingeniería de Telecomunicación)

Alumno:

Grupo:

1. Responde las siguientes cuestiones brevemente (puedes emplear como máximo un folio para todas las preguntas.)
 - a) i) ¿Es siempre posible evitar la recursión? ii) ¿Es siempre aconsejable evitar la recursión? iii) ¿Por qué es ineficiente la solución recursiva de Fibonacci? (0.5 pts.)
 - b) Implementar el algoritmo de búsqueda binaria recursiva en un array. (0.5 pts.)
 - c) ¿Cuáles es la diferencia entre clase y objeto? ¿Cuándo debemos declarar un método en la parte privada de una clase? ¿Es posible declarar constantes en la parte privada de una clase? ¿Cuándo son llamados implícitamente los métodos constructor y el destructor? ¿Tiene sentido llamarlos explícitamente? (0.5 pts.)
 - d) Sean las clases C_1 con métodos m_{11} y m_{12} , y C_2 con métodos m_{21} y m_{22} . El procedimiento P_1 declara los objetos O_1 de la clase C_1 y O_2 de la clase C_2 , respectivamente. Dentro del procedimiento P_1 se llama a otro procedimiento P_2 sin parámetros.
 - i) ¿Sería posible utilizar en el procedimiento P_2 los métodos m_{11} , m_{12} o m_{21} , m_{22} de los objetos O_1 y O_2 ? ii) ¿Si se le pasa al procedimiento P_2 como parámetro un puntero al objeto O_2 podríamos llamar a los métodos m_{21} , m_{22} de O_2 desde P_2 ? ¿Cuál sería la sintaxis, en su caso, de la llamada? iii) Explicar cómo sería posible llamar a los métodos de un objeto de la clase C_1 desde un objeto de la clase C_2 . (0.5 pts.)
2. Definir una estructura de lista doblemente enlazada y circular, como la que se muestra en la figura 1, en la que cada nodo tiene un puntero al siguiente elemento y otro puntero al elemento anterior. Además, al ser circular, el siguiente al último elemento es el primero y el anterior al primer elemento es el último. Los elementos están ordenados ascendentemente. El puntero `ldc` apunta al menor elemento de los que hay en la lista. (3 pts.)

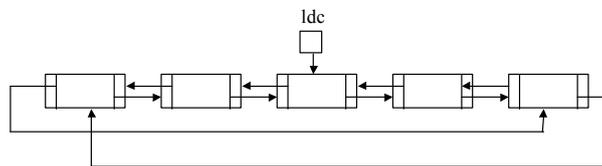


Figura 1: Lista doblemente enlazada circular

Definir, además de la estructura, las siguientes operaciones:

- a) ALGORITMO `Crear(S ListaDobleCircular ldc)`, que crea una lista vacía.
- b) ALGORITMO `B Vacía(E ListaDobleCircular ldc)`, que devuelve VERDADERO si no hay elementos en la lista y FALSO en caso contrario.

- c) ALGORITMO Insertar (ES ListaDobleCircular ldc, E TipoElem elem), que inserta el elemento elem en la lista ldc conservando el orden.
- d) ALGORITMO Borrar (ES ListaDobleCircular ldc, E TipoElem elem), que borra el elemento elem de la lista ldc. Si el elemento no está en la lista, no hace nada.

3. Una *cola de prioridad* es una estructura en la que los elementos que se insertan están ordenados por su prioridad. La prioridad de un elemento se puede obtener a partir del algoritmo ALGORITMO TPrio Prioridad(E TBase b) que se supone que ya está implementado. A la hora de consultar (o extraer) un elemento se consulta (o extrae) el de mayor prioridad.

Implementa la clase Cola de prioridad usando como representación la clase Lista posicional, tal como se definen en los siguientes interfaces de clase. (2 pts.)

INTERFAZ CLASE CColaPrio

TIPOS

TBASE ...

METODOS

Crear()

Destruir()

Encolar(E TBASE x)

Desencolar()

TBASE Frente()

B EstaVacia()

FIN CCola

INTERFAZ CLASE CListaPos

TIPOS

TBASE ...

METODOS

Crear()

Destruir()

Insertar(E N pos, E TBASE x)

Eliminar(E N pos)

TBASE Consultar(E N pos)

N Longitud()

FIN CListaPos

Observa que eres implementador de la clase cola de prioridad y usuario de la clase lista posicional. Enuncia las precondiciones necesarias para las operaciones de la cola de prioridad y tenlas en cuenta a la hora de codificar los métodos.

4. Codificar un algoritmo que tome un árbol binario como parámetro y devuelva su máxima anchura, definida como el mayor de los números de elementos por nivel. Este algoritmo maneja el árbol binario a nivel de usuario. La anchura máxima del árbol de la figura 2 es 4, porque en el nivel 2 hay 4 elementos: 4, 5, 6 y 7. (3 pts.)

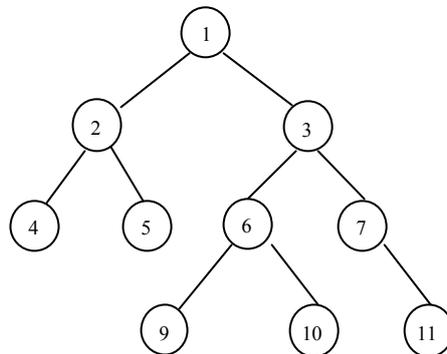


Figura 2: Árbol binario de anchura máxima 4