

Estructuras de Datos (Gestión), curso 2010/11

Práctica única – Ordenación por Tiras

Descripción del Algoritmo

La "Ordenación por Tiras" es un algoritmo de ordenación basado en comparaciones que recibe como entrada una **lista enlazada** de valores que se pueden comparar entre sí (por simplicidad supondremos que son números enteros) y modifica los enlaces entre sus componentes de manera que al terminar la lista se encuentre ordenada de menor a mayor.

La manera de representar la lista se deja a elección del alumno: Simple o doblemente enlazada, lineal o circular, con enlace al primero y/o al último, usando objetos o variables dinámicas de tipo registro, enlaces de tipo dirección de memoria o índices de array... Sin embargo, de entre todas las posibilidades se debe escoger una en la que las operaciones elementales sobre la lista que requiere el algoritmo se realicen de la forma más eficiente posible.

Supondremos que los elementos se encuentran almacenados en **nodos**, representados por registros u objetos que contienen un campo con el valor del elemento y uno o dos campos que apuntan al elemento siguiente y/o anterior.

El algoritmo se puede describir, sucintamente, de la manera siguiente: Si definimos la "tira" de una lista como la sublista formada por su primer elemento y todos los siguientes que forman una secuencia ordenada (por ejemplo, la tira de la lista [3,2,5,7,1,10,8,4] es la sublista [3,5,7,10]) entonces la idea consiste en, usando una lista resultado originalmente vacía, ir **extrayendo** "tiras" de la lista original y **fusionarlas** con la lista resultado, de forma que al final la lista original esté vacía y la lista resultado contenga todos sus elementos originales en orden creciente. Por último, se hace que la lista original sea la lista resultado.

A continuación se describirá en detalle el algoritmo usando un pseudocódigo basado en Pascal, suponiendo que están definidas las siguientes operaciones:

- *esta_vacía(Lista)* : Comprueba si una lista contiene algún elemento.
- *quitar_primero(Lista)* : Devuelve y elimina el primer elemento de la lista.
- *insertar_final(Lista,Nodo)* : Inserta el nodo al final de la lista.
- *nulo(Act)* : Comprueba si el elemento es válido (usado para detectar que se ha llegado al final de una lista)
- *ultimo(Lista)* : Devuelve el último elemento de una lista.
- *quitar(Lista,Nodo)* : Elimina ese nodo de la lista.
- *fusionar(Lista1, Lista2)* : Fusiona los elementos de las dos listas (las cuales están ordenadas), devolviendo la lista fusionada (también ordenada) en *Lista1* y vaciando *Lista2*.

La última operación, *fusionar*, hace uso del algoritmo de **fusión** descrito en el tema 3 (ver transparencias), el cuál se debe adaptar para que actúe sobre listas enlazadas.

```
procedure OrdTiras(var L: TLista);
var
  T : TLista; { Lista temporal que almacena una "tira" }
  R : TLista; { La lista resultado }
  Act,Sig : TNode;
begin
  { T y R inicialmente vacías }
  while not está_vacía(L) do
  begin
    Act := quitar_primero(L)
    insertar_final(T,Act)
    Act := primero(L)
    while not nulo(Act) do
    begin
      Sig := Act.Siguiente
      if Act > ultimo(T) then
      begin
        quitar(L,Act)
        insertar_final(T,Act)
      end;
      Act := Sig
    end; { Se han recorrido todos los elementos de L }
    fusionar(R,T)
  end;
  L := R
end;
```

Requisitos de la Práctica

La práctica consistirá en crear un programa que implemente el algoritmo descrito en el apartado anterior y que se pueda usar para obtener medidas del **número medio de comparaciones**, entre elementos de la lista que se realizan al pedir ordenar un cierto número de listas de **tamaño n** donde los valores de los elementos (se puede suponer que son enteros) se asignan al azar. Con este programa, se deben obtener una serie de datos (número medio de comparaciones según el tamaño) que permitan **estimar**, en notación asintótica, el comportamiento del algoritmo en el caso promedio.

Documentación que se debe entregar

Se deberá entregar el código fuente del programa descrito anteriormente (se pueden usar los lenguajes Eiffel, Java, Free-Pascal o Delphi/Kylix) y un documento (en formato Word, Latex (DVI) o Acrobat PDF) donde se indique lo siguiente:

- Nombre de los alumnos que han realizado la práctica.
- Breve descripción de la representación de las listas, el orden de las operaciones elementales y aspectos relevantes (si los hubiera) de la codificación del algoritmo.
- Una tabla con los valores medidos, con las siguientes columnas: tamaño de la entrada, número de repeticiones para el cálculo de la media, número máximo, medio y mínimo de comparaciones.
- La estimación de la eficiencia en el caso promedio.

Las prácticas se enviarán por **correo electrónico** (cvaca@infor.uva.es) al profesor de la asignatura, y la fecha límite de presentación es el **20 de enero de 2011**.