



ALGORITMOS Y ESTRUCTURAS DE DATOS I

- Junio 1999 -

PARTE I

Dpto. Ing.Electrónica, Sistemas Informáticos y Automática
E.P.S. La Rábida (Universidad de Huelva)

PROBLEMA 1 (6 puntos)

Disponemos de un fichero de acceso secuencia) llamado DATOS.DAT, con información de usuarios de Internet, donde aparece el nombre y la dirección electrónica de correo (e-mail). Dicho fichero contiene email duplicados.

Apartado A (4 puntos):

Se trata de diseñar una acción que procese DATOS.DAT y genere dos fichero de acceso secuencia): uno llamado SIN.DAT sin e-mail duplicados y otro llamado DUPLI.DAT con los e-mail duplicados.

accion duplicados(nombre1, nombre2, nombre3 : cad);

Nombre), nombre2 y nombre3 se refieren a los nombres de los ficheros DATOS.DAT, SIN.DAT y DUPLI.DAT respectivamente.

Debido al tamaño del fichero no se puede utilizar estructuras del tipo tabla, ni asignación dinámica de memoria, para almacenar los registros del mismo.

La estructura de los registros es

TipoReg = registro
Nombre: Cadena;
E_mail: Cadena;
fregistro;

Cadena = tabla[1..12] de carácter;

Apartado B (2 puntos):

Suponiendo que el tamaño del fichero DATOS.DAT lo permite, diseñar una acción que nos devuelva los duplicados del fichero DATOS.DAT en una lista doblemente enlazada (desordenada) en la cual las inserciones se realicen por el final.

accion DUPLICADOS (nombre1,nombre2:Cadena)

Estructura de la lista

punt=puntero a nodol;
nodol=registro
Nombre: Cadena;
e_mail: Cadena;
sig,ant:punt;
fregistro

Utilizar técnicas de Diseño Descendente



ALGORITMOS Y ESTRUCTURAS DE DATOS I

- Junio 1999 -

PARTE II

Dpto. Ing. Electrónica, Sistemas Informáticos y Automática
E.P.S. La Rábida (Universidad de Huelva)

PROBLEMA 2 (2 puntos)

INTRODUCCION A LA RECURSIVIDAD

Aplicación a la combinatoria. - Suponga que tenemos "m" elementos y pretendemos ocupar "n" lugares distintos con ellos.

Las distintas formas que tengo de hacerlo reciben el nombre siguiente:

- Variaciones con repetición de "m" elementos tomados "n" a "n", si cada elemento puede ocupar más de un lugar, y su fórmula es:

$$VR_{m,n} = m^n$$

- Variaciones de "m" elementos tomados "n" a "n", si cada elemento puede ocupar un lugar a lo sumo, y su fórmula es:

$$V_{m,n} = m * (m-1) * \dots * (m-n+1)$$

Pregunta. - Desarrolle la siguiente función recursiva:

función variaciones (tipovar: carácter; m, n : entero) : entero;

(* tipovar = "s" si es variaciones sin repetición.

= "c" si es variaciones con repetición.

m= número de elementos.

n = número de lugares.

valor devuelto = número de variaciones (con o sin repetición).

*)

PROBLEMA 3 (1 punto)

ALGORITMOS DE ORDENACION

Suponga que un compañero suyo le proporciona la siguiente acción para ordenar la tabla "TA" entre los índices "inicio" y "fin":

tipo T = tabla[1.. 100] de entero.

acción desconocida (var TA: T; inicio,fin :entero);

var

k, valor, i, izq, dcha : entero

fvar

inicio

para i:= inicio + 1 hasta fin hacer

inicio

valor := TA[i];

izq := inicio;

dcha := i -1;

mientras izq <= dcha hacer

inicio

k:= (izq + dcha) div 2;

si TA[i] < TA[k] → dcha := k-1

□ TA[i] >= TA[k] → izq := k+1

fin

para k := i-1 hasta izq cada -1 hacer
 TA[k+1] := TA[k];

TA[izq]:=valor;

fin;

faccion

Pregunta.- Haga un seguimiento claro de la siguiente llamada *desconocida(T1,2,5)* siendo T1 :

1	2	3	4	5	6	7
10	6	4	9	5	1	8

Explicando a continuación el motivo por el que se puede decir si es básicamente una ordenación por el sistema de la burbuja, selección, inserción o shell.

PROBLEMA 4 (1 punto)

La siguiente acción fue diseñada con el fin de eliminar todos los duplicados de una lista L (cada nodo de la lista puede tener más de un duplicado). ¿Por qué no funciona siempre y cómo se podría arreglar?

acción ELIM (var L:LISTA);

var

p,q:posición

fvar

inicio

p:=PRIMERO(L);

mientras p<>NULO hacer

inicio

q:=SIGUIENTE(p,L);

mientras q<>NULO hacer

si igual(RECUPERA(p,L),RECUPERA(q,L)) → SUPRIMIR (q,L)

□ no(igual(RECUPERA(p,L),RECUPERA(q,L))) → q:=SIGUIENTE (q,L);

...

fin;

faccion

NOTA 1:

tipo

LISTA=puntero a nodo;

nodo=registro

elemento:tipo_elemento;

siguiente:LISTA

fregistro

ftipo

var L:LISTA; p,q,r:posicion fvar

NOTA 2:

PRIMERO (L): Devuelve la primera posición de la lista L.

SIGUIENTE(p,L). Devuelve la posición siguiente a p en la lista L.

RECUPERA(p,L): Devuelve el elemento que está en la posición p de la lista L.

SUPRIMIR(p,L): Elimina el elemento en la posición p de la lista L.