Fundamentos de la programación

7A Más sobre ordenación

ANEXO

Grado en Ingeniería Informática
Grado en Ingeniería del Software
Grado en Ingeniería de Computadores

Luis Hernández Yáñez
Facultad de Informática
Universidad Complutense

Índice

Luis Hernández Yáñez

Ordenación por intercambio

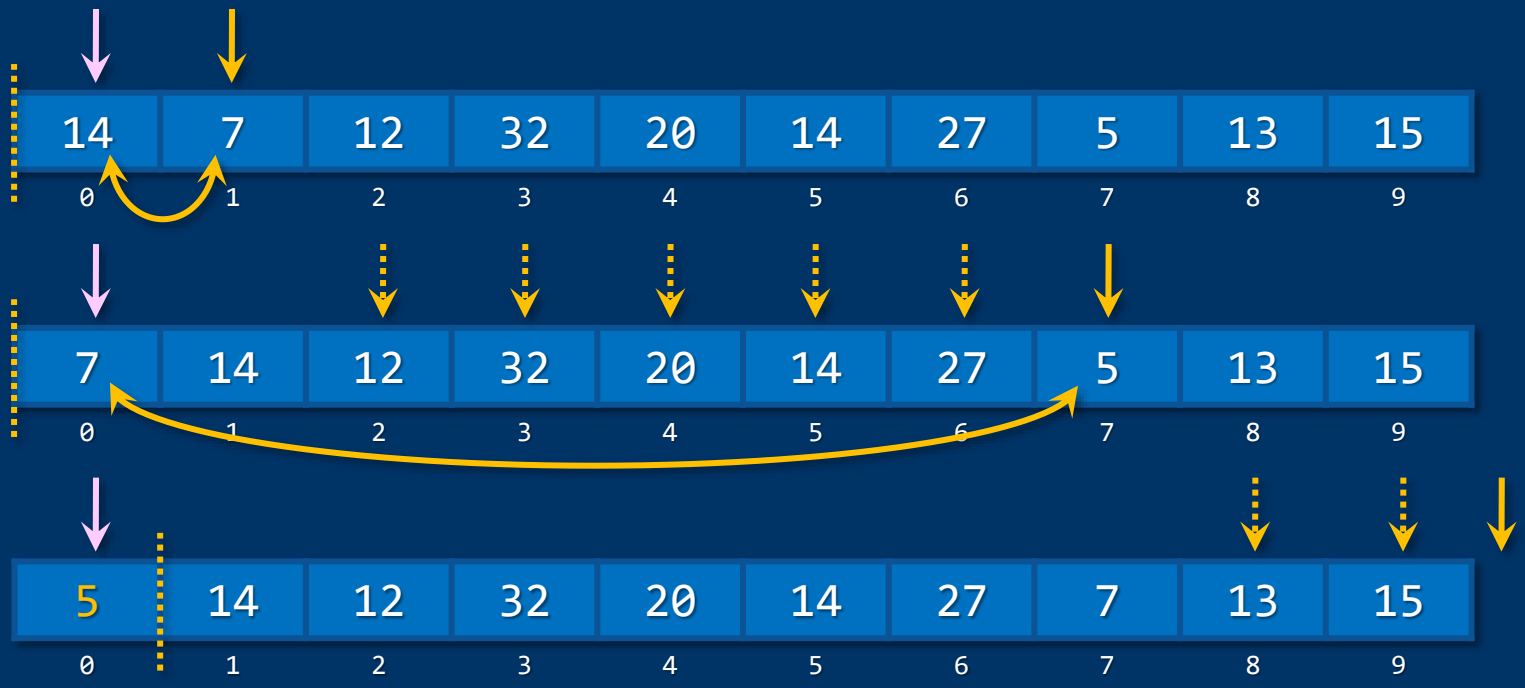
Luis Hernández Yáñez

Ordenación por intercambio

Algoritmo de ordenación por intercambio

Variación del método de selección directa

Se intercambia el elemento de la posición que se trata en cada momento siempre que se encuentra uno que es menor:



Luis Hernández Yáñez

Ordenación por intercambio

intercambio.cpp

```
const int N = 10;
typedef int tLista[N];
tLista lista;
...
for (int i = 0; i < N - 1; i++) {
// Desde el primer elemento hasta el penúltimo
   for (int j = i + 1; j < N; j++) {
   // Desde i+1 hasta el final
      if (lista[j] < lista[i]) {
         int tmp;
         tmp = lista[i];
         lista[i] = lista[j];
         lista[j] = tmp;
      }
   }
}
```

Igual número de comparaciones, muchos más intercambios

No es estable

Luis Hernández Yáñez

Mezcla de dos listas ordenadas

Luis Hernández Yáñez

Mezcla de listas ordenadas

Mezcla de dos listas ordenadas en arrays

```
const int N = 100;
typedef struct {
  int elementos[N];
  int cont;
} tLista;
```

Un índice para cada lista, inicializados a 0 (principio de las listas)

Mientras que no lleguemos al final de alguna de las dos listas:

Elegimos el elemento menor de los que tienen los índices

Lo copiamos en la lista resultado y avanzamos su índice una posición

Copiamos en la lista resultado los que queden en la lista no acabada

Mezcla de listas ordenadas

```
void mezcla(tLista lista1, tLista lista2, tLista &listaM) {
   int pos1 = 0, pos2 = 0;
   listaM.cont = 0;

   while ((pos1 < lista1.cont) && (pos2 < lista2.cont)
          && (listaM.cont < N)) {
      if (lista1.elementos[pos1] < lista2.elementos[pos2]) {
         listaM.elementos[listaM.cont] = lista1.elementos[pos1];
         pos1++;
      }
      else {
         listaM.elementos[listaM.cont] = lista2.elementos[pos2];
         pos2++;
      }
      listaM.cont++;
   }
   ...
```

Luis Hernández Yáñez

Mezcla de listas ordenadas

mezcla1.cpp

```
    // Pueden quedar datos en alguna de las listas
    if (pos1 < lista1.cont) {
        while ((pos1 < lista1.cont) && (listaM.cont < N)) {
            listaM.elementos[listaM.cont] = lista1.elementos[pos1];
            pos1++;
            listaM.cont++;
        }
    }
    else { // pos2 < lista2.cont
        while ((pos2 < lista2.cont) && (listaM.cont < N)) {
            listaM.elementos[listaM.cont] = lista2.elementos[pos2];
            pos2++;
            listaM.cont++;
        }
    }
}
```

```
Primera lista:
1   4   5   8   12   12   15   18   24   31   45   49   63
Segunda lista:
2   3   9   14   15   23   28   42   58   73   79   84   88   93
Lista con la mezcla:
1   2   3   4   5   8   9   12   12   14   15   15   18   23   24   28   31   42
   45   49   58   63   73   79   84   88   93
```

Luis Hernández Yáñez

Mezcla de listas ordenadas

Mezcla de dos listas ordenadas en archivos

```
void mezcla(string nombre1, string nombre2, string nombreM) {
// Mezcla las secuencias en los archivos nombnre1 y nombre2
// generando la secuencia mezclada en el archivo nombreM
   ifstream archivo1, archivo2;
   ofstream mezcla;
   int dato1, dato2;

   // Los archivos existen y son correctos
   archivo1.open(nombre1.c_str());
   archivo2.open(nombre2.c_str());
   mezcla.open(nombreM.c_str());
   archivo1 >> dato1;
   archivo2 >> dato2;
   while ((dato1 != -1) && (dato2 != -1)) {
   // Mientras quede algo en ambos archivos
      ...
```

Luis Hernández Yáñez

Mezcla de listas ordenadas

```
        if (dato1 < dato2) {
            mezcla << dato1 << endl;
            archivo1 >> dato1;
        } else {
            mezcla << dato2 << endl;
            archivo2 >> dato2;
        }
    } // Uno de los dos archivos se ha acabado
    if (dato1 != -1) { // Quedan en el primer archivo
        while (dato1 != -1) {
            mezcla << dato1 << endl;
            archivo1 >> dato1;
        }
    }
    else { // Quedan en el segundo archivo
        while (dato2 != -1) {
            mezcla << dato2 << endl;
            archivo2 >> dato2;
        }
    }
    ...
```

Luis Hernández Yáñez

Mezcla de listas ordenadas

mezcla2.cpp

```
    archivo2.close();
    archivo1.close();
    mezcla << -1 << endl;
    mezcla.close();
}
```

Luis Hernández Yáñez

Acerca de *Creative Commons*

Licencia CC (Creative Commons)

Este tipo de licencias ofrecen algunos derechos a terceras personas bajo ciertas condiciones.

Este documento tiene establecidas las siguientes:

- Reconocimiento (*Attribution*):
  En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
- No comercial (*Non commercial*):
  La explotación de la obra queda limitada a usos no comerciales.
- Compartir igual (*Share alike*):
  La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Pulsa en la imagen de arriba a la derecha para saber más.

Luis Hernández Yáñez