**Programa 5, Ordenamiento Burbuja y Burbuja Mejorada**

Para el desarrollo de estos dos programas, se tuvo que hacer una pequeña investigación sobre estructuras y cadenas.

La manera en que se generaron las 30000 cadenas con números aleatorios fue la siguiente:

Primero definir una estructura que tuviera una cadena de 30 caracteres, después de hacer eso podríamos decir que hicimos un arreglo de estructuras de manera que así se genero el espacio en memoria que es necesario para guardar todos los dígitos de las 30000 cadenas.

Luego se utilizo un proceso iterativo para acceder a cada localidad del arreglo de estructuras y a su vez ir recorriendo la cadena de caracteres uno a uno para asignar valores aleatorios, utilizando el código UNICODE de los dígitos 0 al 9 mediante la función random que la sintaxis que se utilizo en el programa fue la siguiente:

cadena[i] = '0'+ ( rand() % ('0' - '9') );

Después de haber generado las 30000 cadenas aleatorias descritas con el procedimiento anterior, se procedió a ordenarlas según la opción seleccionada, para esto se tuvo que utilizar algunas funciones de la librería String que nos sirve para la manipulación de cadenas:

* strcmp(cadena1,cadena2): Esta función compara dos cadenas, necesita de parámetros dos cadenas, y devuelve un valor entero, si la cadena1 es menor que la cadena2 devuelve un numero menor a 0, si son idénticas devuelve el cero y si la cadena1 es mayor que la cadena2 entonces devuelve un numero mayor a 0.
* Strcpy(cadenaDest,cadenaSource): Esta función nos permite copiar los elementos de una cadena a otra, sus parámetros de entradas son dos cadenas y de salida es la cadenaDest o la cadena a la cual se copio la otra cadena.

Con estas dos funciones se logro hacer el ordenamiento, determinar que cadena era mayor que la otra nos sirvió strcmp y strcpy nos sirvió para hacer el intercambio. Al igual dependiendo de la opción seleccionada por el usuario.

También al ejecutar el proceso de depuración y pruebas de escritorio se analizo como funciona este tipo de ordenamiento, como funciona su proceso de ordenación, a manera de explicación se hará un esquema breve de lo comprendido de este método y su análisis de ventajas y desventajas.

**Ordenamiento Burbuja**

Para explicar el método supongamos que tenemos un arreglo de enteros que tiene los siguientes elementos de la siguiente manera

5—0 4—1 3—2 1—3 2—4 6—5

Donde la posición es el segundo numero.

La manera de funcionamiento del método seria la siguiente compararíamos el numero que se encuentra en la posición 0, con todos los demás números para ver si se cumple la condición en este caso lo ordenaremos de menor a mayor. Entonces el primer paso será comparar el valor de la posición 0, con el de la posición 1.

Entonces decimos 5 es mayor que 4? Debido a que si se hace el intercambio y quedaría como:

4—0 5—1 3—2 1—3 2—4 6—5

Ahora comparamos el número que se encuentra en la posición 1 con el de la posición 2, por lo que decimos 5 es mayor a 3? Debido a que si se hace el intercambio

4—0 3—1 5—2 1—3 2—4 6—5

Ahora comparamos el número que se encuentra en la posición 2 con el de la posición 3, por lo que decimos 5 es mayor que 1? Debido a que si se hace el intercambio

4—0 3—1 1—2 5—3 2—4 6—5

Ahora comparamos el número que se encuentra en la posición 3 con el de la posición 4, por lo que decimos 5 es mayor que 2? Debido a que si se hace el intercambio

4—0 3—1 1—2 2—3 5—4 6—5

Esta seria la última comparación que se debe hacer, por lo que comparamos el número que esta en la posición 4 con el de posición 5, por lo que decimos 5 es mayor a 6? Debido a que no, no se realiza el intercambio y queda así.

Se vuelve a repetir el proceso comparando el numero de la posición 0 con el de posición 1, y se repetirá dicho proceso hasta que el arreglo quede ordenado de menor a mayor. Al finalizar este proceso quedaría de la siguiente manera:

1—0 2—1 3—2 4—3 5—4 6—5

La ventaja es que ordena todos los datos, de una manera sencilla de comprender y fácil de analizar, la desventaja es que compara todos los números en cada iteración estén o no estén ordenados lo que lo hace un poco ineficiente y tardado.

**Ordenamiento Burbuja Mejorado**

Para explicar el método supongamos que tenemos un arreglo de enteros que tiene los siguientes elementos de la siguiente manera

5—0 4—1 3—2 1—3 2—4 6—5

Donde la posición es el segundo numero.

La manera de funcionamiento del método seria la siguiente compararíamos el numero que se encuentra en la posición 0, con todos los demás números para ver si se cumple la condición en este caso lo ordenaremos de menor a mayor. Entonces el primer paso será comparar el valor de la posición 0, con el de la posición 1.

Entonces decimos 5 es mayor que 4? Debido a que si se hace el intercambio y quedaría como:

4—0 5—1 3—2 1—3 2—4 6—5

Ahora comparamos el número que se encuentra en la posición 1 con el de la posición 2, por lo que decimos 5 es mayor a 3? Debido a que si se hace el intercambio

4—0 3—1 5—2 1—3 2—4 6—5

Ahora comparamos el número que se encuentra en la posición 2 con el de la posición 3, por lo que decimos 5 es mayor que 1? Debido a que si se hace el intercambio

4—0 3—1 1—2 5—3 2—4 6—5

Ahora comparamos el número que se encuentra en la posición 3 con el de la posición 4, por lo que decimos 5 es mayor que 2? Debido a que si se hace el intercambio

4—0 3—1 1—2 2—3 5—4 6—5

Esta seria la última comparación que se debe hacer, por lo que comparamos el número que esta en la posición 4 con el de posición 5, por lo que decimos 5 es mayor a 6? Debido a que no, no se realiza el intercambio y queda así.

Aquí es donde diferirá esta modificación del método a diferencia del método anterior ya que este método deja de comparar los datos de uno en uno según la iteración por ejemplo en la primera iteración compara n elementos pero para la segunda iteración compara n-1 elementos y así hasta que queda ordenado.

Lo interesante de este programa es la aplicación, ya que se debe analizar que al ser una cadena de texto, no estamos comparando en si el valor numérico de los dígitos, sino más bien su valor en código ya sea ASCII o UNICODE. Es lo que nos permite ordenarlos de menor a mayor, y de la misma manera se pueden ordenar nombres de la “a” a “z”, considerando que la letra “a” tiene el valor mas pequeño dentro del código UNICODE y la “z” el mayor. Como se puede observar en la ilustración 1.

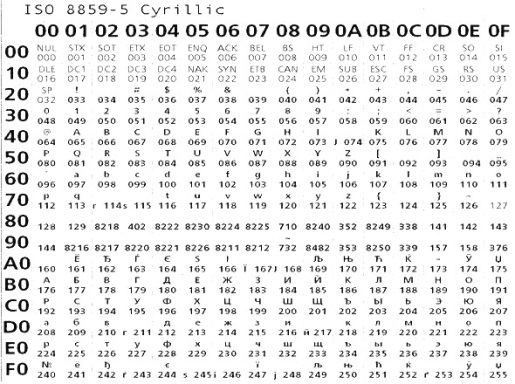


Ilustración 1 Tabla de código UNICODE obtenida de la página de IBM

Referencias Electrónicas:

<http://msdn.microsoft.com/en-us/library/e0z9k731.aspx>

<http://msdn.microsoft.com/en-us/library/kk6xf663.aspx>

<http://www.ibm.com/developerworks/linux/library/l-u-cyr/>