**Ordenamiento de Burbuja**

El primer ordenamiento que presentamos es quizá el más ampliamente conocido entre los estudiantes que se inician en la programación. Una de las características de este ordenamiento es que es fácil de entender y programar. Aunque, es uno de los algoritmos mas ineficiente.

La idea básica subyacente en el ordenamiento de burbuja es pasar a través del arreglo de datos varias veces en forma secuencial. Cada paso consiste en la comparación de cada elemento en el arreglo con su sucesor (x[i] con x[i+1]) y el intercambio de los dos elementos si no están en el orden correcto. Considérese el siguiente arreglo de datos:

 25 57 48 37 12 92 86 33

En el primer paso se realizan las siguientes operaciones:

x[0] con x[1] (25 con 57) no intercambio.

x[1] con x[2] (57 con 48) intercambio.

x[2] con x[3] (57 con 32) intercambio.

x[3] con x[4] (57 con 12) intercambio.

x[4] con x[5] (57 con 92) no intercambio.

x[5] con x[6] (92 con 86) intercambio.

x[6] con x[7] (92 con 33) intercambio.

Así después del primer paso, el arreglo está en el siguiente orden:

25 48 37 12 57 86 33 92

Obsérvese que después del primer paso, el elemento mayor (en este caso 92) está en la posición correcta dentro del arreglo. En general x[n-i] estará en su posición correcta después de la iteración i. El método se lama ordenamiento de burbuja porque cada número “burbujea” con lentitud hacia su posición correcta. El conjunto completo de iteraciones es:

iteración 0 : 25 57 48 37 12 92 86 33

iteración 1: 25 48 37 12 57 86 33 92

iteración 2: 25 37 12 48 57 33 86 92

iteración 3: 25 12 37 48 33 57 86 92

iteración 4: 12 25 37 33 48 57 89 92

iteración 5: 12 25 33 37 48 57 89 92

La implementación de este algoritmo en Java queda como:

  void Burbuja(int x[], int n)

{

int b, j, t;

do

{

b = 0;

for(j=0; j<n; j++)

{

if(x[j] > x[j+1])

{

t = x[j];

x[j] = x[j+1];

x[j+1] = t;

b++;

}

}

n--;

}

while(b > 0);

}