

Ordenamiento en C/C++

Mag. Ing. Nancy López

Ordenamiento

- Ordenar es reorganizar una secuencia de datos de una manera determinada.
- Por ejemplo en la guía telefónica, tenemos nombres, teléfonos y direcciones.
- En distintas secciones puede estar ordenado por nombre, por calle o por número telefónico.
- Se toma un elemento como clave y los demás datos no se toman en consideración pero sí se reordenan pero siempre teniendo en cuenta el campo clave.

Ordenamiento – Bubble Sort

```
for(i=0; i<TAM; i++)  
    for(j=0; j<TAM-1; j++)  
        if( clave j < clave j+1 ) //o clave j > clave j+1  
        {  
            //intercambio  
        }
```

Ordenamiento – Bubble Sort

- Análisis del algoritmo:
- La ordenación de burbuja tanto en el caso medio como en el peor de los casos utiliza aproximadamente $n^2/2$ comparaciones y $n^2/2$ intercambios, por lo cual
- $T(n) = O(n^2)$.

Ordenamiento – Bubble Sort

Ejemplo: `int datos[6]={40,21,4,9,10,35};`

Primera pasada:

{40, 21, 4, 9, 10, 35}

{21, 40, 4, 9, 10, 35} <-- Se cambia el 21 por el 40.

{21, 4, 40, 9, 10, 35} <-- Se cambia el 40 por el 4.

{21, 4, 9, 40, 10, 35} <-- Se cambia el 9 por el 40.

{21, 4, 9, 10, 40, 35} <-- Se cambia el 40 por el 10.

{21, 4, 9, 10, 35, 40} <-- Se cambia el 35 por el 40.

Segunda pasada:

{21, 4, 9, 10, 35, 40}

{4, 21, 9, 10, 35, 40} <-- Se cambia el 21 por el 4.

{4, 9, 21, 10, 35, 40} <-- Se cambia el 9 por el 21.

{4, 9, 10, 21, 35, 40} <-- Se cambia el 21 por el 10.

{4, 9, 10, 21, 35, 40} <-- No hay cambio entre el 21 y el 35.

{4, 9, 10, 21, 35, 40} <-- El 35 y el 40, no se comparan, porque el 40 es el mayor.

En este ejemplo, los datos ya están ordenados, pero para comprobarlo habría que hacer una tercera, cuarta y quinta comprobación.

Bubble Sort o Burbuja

```
const int TAM=100;
int main()
{
    int vector[TAM], i,j, aux;
    //Cargar el vector
    for(i=0; i<TAM;i++)
        for(j=0;j<TAM-1;j++)
            if(vector[j]<vector[j+1])
            {
                aux=vector[j];
                vector[j]=vector[j+1];
                vector[j+1]=aux;
            }
    //mostrar el vector.
}
```

**Para vectores de una
dimensión**

Ordena el vector
de mayor a menor.
aux tiene que ser
del mismo tipo de
dato que se va a
intercambiar.

Bubble Sort o Burbuja

```
for(i=0; i<FIL; i++)  
  for(j=0; j<FIL-1; j++)  
    if(vector[j][0]<vector[j+1][0])  
    {  
      for(k=0; k<COL; k++)  
      {  
        aux=vector[j][k];  
        vector[j][k]=vector[j+1][k];  
        vector[j+1][k]=aux;  
      }  
    }  
}
```

Para vectores de dos dimensiones (matrices)

Ordena el vector de mayor a menor tomando como clave la primer columna.

Bubble Sort o Burbuja

```
const int FIL=6;  
const int CAR=20;
```

```
int i,j;  
char nombres[FIL][CAR], aux[CAR];
```

```
for(i=0; i<FIL;i++)  
    for(j=0;j<FIL-1;j++)  
        if(strcmp(nombres[j],nombres[j+1])>0)  
        {  
            strcpy(aux,nombres[j]);  
            strcpy(nombres[j],nombres[j+1]);  
            strcpy(nombres[j+1],aux);  
        }
```

**Para cadenas de
caracteres**

Ordena las
cadenas de la A
a la Z.

Bubble Sort o Burbuja

- Si se tiene una matriz con cadenas relacionada con otra matriz de números (alumnos y notas por materias, por ej.)
 1. Si se ordena por nombre, la clave nombre se considera en el if (usando strcmp) y en el intercambio se **intercambian los nombres y también las notas.**
 2. Si se ordena por un criterio numérico, se considera ese criterio en el if (usando < o >) y en el intercambio se **intercambian los nombres y también las notas.**

Ejemplos en Internet

- <https://www.youtube.com/watch?v=lyZQPjUT5B4>
- https://www.youtube.com/watch?v=Cq7SMsQBEUw&list=PLZh3kxyHrVp_AcOanN_jpuQbcMVdXbqei&index=7