



Seminario de Algoritmia

REPORTE DE PRÁCTICA

IDENTIFICACIÓN DE LA PRÁCTICA

Práctica	2	Nombre de la práctica	Ordenamiento burbuja
Fecha	02/09/21	Nombre del profesor	Alma Nayeli Rodríguez Vázquez
Nombre de los integrantes del equipo		1. Cárdenas Pérez Calvin Cristopher	
		2. Farfán de León José Osvaldo	
		3. García Martínez Noe Aaron	

OBJETIVO

El objetivo de esta práctica consiste en implementar el algoritmo de ordenamiento burbuja.

PROCEDIMIENTO

Realiza la implementación siguiendo estas instrucciones.

Implementa el algoritmo de ordenamiento burbuja utilizando Matlab/Octave y C++ / Python. Para la implementación, utiliza un vector de 10 elementos enteros con valores entre 1 y 100. Apóyate en el siguiente algoritmo:

```
ALGORITHM BubbleSort( $A[0..n-1]$ )  
  //Sorts a given array by bubble sort  
  //Input: An array  $A[0..n-1]$  of orderable elements  
  //Output: Array  $A[0..n-1]$  sorted in nondecreasing order  
  for  $i \leftarrow 0$  to  $n-2$  do  
    for  $j \leftarrow 0$  to  $n-2-i$  do  
      if  $A[j+1] < A[j]$  swap  $A[j]$  and  $A[j+1]$ 
```



Seminario de Algoritmia

IMPLEMENTACIÓN

Agrega el código de tu implementación aquí.

```
A=randi(100,1,10);  
A  
n=numel(A);  
for i=1:n-1  
    for j=1:n-i  
        if A(j+1)<A(j)  
            aux=A(j);  
            A(j)=A(j+1);  
            A(j+1)=aux;  
        End  
    End  
End  
A
```

Código de Matlab/Octave

```
#include <iostream>  
using std::cout;  
using std::endl;  
#include <ctime>  
const int tamano = 10, MAXIMO = 100;  
int i,j,aux;  
int main(){  
  
    int arreglo[ tamano ] = { 0 };  
    srand( time( 0 ) );  
    for ( i = 0; i < tamano; ++i )  
        arreglo[ i ] = rand() % MAXIMO;  
    cout << "Los valores del arreglo son:\n";  
    for ( i = 0; i < tamano; ++i ){  
        cout <<"arreglo[" << i << "]" = " << arreglo[ i ];  
        cout << "\n";  
    }  
    for(i=0; i<tamano; ++i){  
        for(j=0; j<tamano; ++j){  
            if(arreglo[j] > arreglo[j+1]){  
                aux = arreglo[j];  
                arreglo[j] = arreglo[j+1];  
                arreglo[j+1] = aux;  
            }  
        }  
    }  
    cout << "Los valores del arreglo ordenados por metodo burbuja  
son:\n";  
    for ( i = 0; i < tamano; ++i ){  
        cout <<"arreglo[" << i << "]" = " << arreglo[ i ];  
        cout << "\n";  
    }  
}
```



Seminario de Algoritmia

```
}  
}
```

Código en C++/Python

RESULTADOS

Agrega la imagen de la consola con el despliegue de los resultados obtenidos.

The screenshot shows the OctaveOnline interface. On the left, a 'Variables' panel lists: [1x10] A, # ans, # aux, # i, # j, # n. The main console area displays the following code and its output:

```
octave:1> A=randi(100,1,10);  
octave:2> A  
A =  
    65    54    10    41    31    98    34    66    84    77  
  
octave:3> n=numel(A);  
octave:4> for i=1:n-1  
>> for j=1:n-i  
>> if A(j+1)<A(j)  
>> aux=A(j);  
>> A(j)=A(j+1);  
>> A(j+1)=aux;  
>> end  
>> A  
>> end  
>> end  
=  
    54    65    10    41    31    98    34    66    84    77  
  
A =  
  
    54    10    65    41    31    98    34    66    84    77  
  
A =  
  
    54    10    41    65    31    98    34    66    84    77  
  
A =
```



Seminario de Algoritmia

```
OctaveOnline
Variables
[1x10] A
# ans
# aux
# i
# j
# n

54 10 41 65 31 98 34 66 84 77
A =
54 10 41 31 65 98 34 66 84 77
A =
54 10 41 31 65 98 34 66 84 77
A =
54 10 41 31 65 34 98 66 84 77
=
54 10 41 31 65 34 66 98 84 77
=
54 10 41 31 65 34 66 84 98 77
A =
54 10 41 31 65 34 66 84 77 98
A =
10 54 41 31 65 34 66 84 77 98
A =
```

```
OctaveOnline
Variables
10 54 41 31 65 34 66 84 77 98
[1x10] A
# ans
# aux
# i
# j
# n

A =
10 41 54 31 65 34 66 84 77 98
A =
10 41 31 54 65 34 66 84 77 98
A =
10 41 31 54 65 34 66 84 77 98
A =
10 41 31 54 34 65 66 84 77 98
A =
10 41 31 54 34 65 66 84 77 98
A =
10 41 31 54 34 65 66 77 84 98
A =
```



Seminario de Algoritmia

```
OctaveOnline  
Variables  
[1x10] A  
# ans  
# aux  
# i  
# j  
# n  
A =  
10 31 41 54 34 65 66 77 84 98  
A =  
10 31 41 54 34 65 66 77 84 98  
A =  
10 31 41 34 54 65 66 77 84 98  
A =  
10 31 41 34 54 65 66 77 84 98  
=  
10 31 41 34 54 65 66 77 84 98  
=  
10 31 41 34 54 65 66 77 84 98  
A =  
10 31 41 34 54 65 66 77 84 98  
A =  
10 31 41 34 54 65 66 77 84 98
```

```
octave-online.net/#cmd=A%3Drandi(100%2C1%2C10)%3B  
OctaveOnline  
Variables  
[1x10] A  
# ans  
# aux  
# i  
# j  
# n  
A =  
10 31 34 41 54 65 66 77 84 98  
A =  
10 31 34 41 54 65 66 77 84 98  
A =  
10 31 34 41 54 65 66 77 84 98  
A =  
10 31 34 41 54 65 66 77 84 98  
A =  
10 31 34 41 54 65 66 77 84 98  
A =  
10 31 34 41 54 65 66 77 84 98  
octave:5> A  
A =  
10 31 34 41 54 65 66 77 84 98  
NOTICE: Due to inactivity, your session will expire in five minutes.
```

Resultados Matlab/Octave



Seminario de Algoritmia

```
Los valores del arreglo son:
arreglo[0] = 42
arreglo[1] = 58
arreglo[2] = 95
arreglo[3] = 95
arreglo[4] = 35
arreglo[5] = 55
arreglo[6] = 33
arreglo[7] = 87
arreglo[8] = 30
arreglo[9] = 69
Los valores del arreglo ordenados por metodo burbuja son:
arreglo[0] = 16
arreglo[1] = 30
arreglo[2] = 33
arreglo[3] = 35
arreglo[4] = 42
arreglo[5] = 55
arreglo[6] = 58
arreglo[7] = 69
arreglo[8] = 87
arreglo[9] = 95
Process returned 0 (0x0)   execution time : 0.081 s
Press any key to continue.
```

Resultados C++/Python

CONCLUSIONES

Escribe tus observaciones y conclusiones.

Noé Aaron García Martínez:

El algoritmo burbuja es un algoritmo de ordenamiento sencillo de implementar, de la misma manera que el algoritmo anterior no es necesario generar otro arreglo o lista de datos para que el algoritmo funcione, solamente se requiere de una variable auxiliar.

La sencillez de este algoritmo también causa que no sea tan eficiente.

El algoritmo Burbuja hace el ordenamiento mediante comparaciones de datos en donde se toma un primer valor, si el valor es mayor al que está en el siguiente índice, se cambian de lugar, de lo contrario se toma el valor en el siguiente índice como el valor para comparar, este proceso se repite hasta que se tiene el valor más grande en el último índice del arreglo. Decimos que este valor ya está ordenado y se toma otro en el inicio del arreglo asegurándose de no comparar con los valores ya ordenados.

Cardenas Pérez Calvin Cristopher:

Fue un método fácil de implementar y además me gusto ver cómo trabaja este metodo en dos programas distintos, hablando de Matlab y de c++, pero a pesar de lo simple que fue hacerlo y su eficacia, hay métodos mas rapidos y que consumen menos tiempo de computadora.

Farfán de León José Osvaldo:

En este metodo fue muy sencillo el implementar ya que consiste en ir recorriendo cada uno de los indices del arreglo e ir subiendo o moviendo de casilla los valores asta que se encontraba a un numero mayor al que se estaba moviendo en caso de encontrar a un numero menor intercambiaban la posicion y el numero que se estaba moviendo continua subiendo es hay de donde le llaman metodo burbuja los numeros grande suben de uno a uno asta que todos los numeros se ordenan de la manera correcta