



Facultad de Estudios Estadísticos

Grado en Estadística Aplicada

Programación II - 801587

Curso 2019-2020

Arrays Unidimensionales. Ordenación, búsqueda y mezcla.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70



INDICE

Arrays Unidimensionales:

Busqueda

- Sin ordenar. Secuencial
- Ordenados. Secuencial y Bipartición

Ordenación

- Burbuja, Selección, Inserción, Quick Sort

Mezcla

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70



Búsqueda

Sin ordenar. Secuencial

```
#include <iostream>
using namespace std;

const int MAX=10;
typedef int tArraySinOrdenar[MAX];

int BusquedaSinOrdenar(int a[], int numero)
{
    bool encontrado = false;
    int indice = 0;
    while( (indice < MAX) && (encontrado==false) )
    {
        if ( a[indice]==numero ) encontrado=true;
        indice++;
    }
    if(encontrado) return indice; else return -1;
    //Podriamos devolver bool true si lo encuentra false en caso contrario
}

int main() {

    tArraySinOrdenar ASO = {2,4,1,6,3,7,5,9,0,8};

    int numeroBuscar = 5;

    int resultado = BusquedaSinOrdenar(ASO, numeroBuscar);

    if (resultado==-1)
        cout << "No encontrado";
    else
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Búsqueda

Ordenado. Secuencial

```
#include <iostream>
using namespace std;

const int MAX=10;
typedef int tArrayOrdenado[MAX];

int BusquedaOrdenado(int a[], int numero)
{
    bool encontrado = false;
    int indice = 0;
    while( (indice < MAX) && (numero >= a[indice]) && (encontrado==false) )
    {
        if ( a[indice]==numero ) encontrado=true;
        indice++;
    }
    if(encontrado) return indice; else return -1;
    //Podriamos devolver bool true si lo encuentra false en caso contrario
}

int main() {

    tArrayOrdenado AO = {1,2,3,4,5,6,7,8,9,10};

    int numeroBuscar = 5;

    int resultado = BusquedaOrdenado(AO, numeroBuscar);

    if (resultado==-1)
        cout << "No encontrado";
    else
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



Búsqueda

Comparativa: Sin ordenar y Ordenado. Secuencial

```
int BusquedaSinOrdenar(int a[], int numero)
{
    bool encontrado = false;
    int indice = 0;
    while( (indice < MAX) && (encontrado==false) )
    {
        if ( a[indice]==numero ) encontrado=true;
        indice++;
    }
    if(encontrado) return indice; else return -1;
    //Podriamos devolver bool true si lo encuentra false en caso contrario
}

int BusquedaOrdenado(int a[], int numero)
{
    bool encontrado = false;
    int indice = 0;
    while( (indice < MAX) && (numero >= a[indice]) && (encontrado==false) )
    {
        if ( a[indice]==numero ) encontrado=true;
        indice++;
    }
    if(encontrado) return indice; else return -1;
    //Podriamos devolver bool true si lo encuentra false en caso contrario
}
```

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Búsqueda

Ordenado. Bipartición.

```
int BusquedaBiparticion(int a[], int numero)
{
    bool encontrado = false;
    int izquierda = 0;
    int derecha = MAX;
    int centro;

    while( (izquierda <= derecha) && (encontrado==false) )
    {
        centro = (izquierda+derecha) / 2 ;
        cout << " En la posicion " << centro << " esta el valor " << a[centro];
        //Para seguimiento
        if (a[centro]==numero) {
            encontrado = true;
        }
        else
        {
            if (a[centro] > numero )
                derecha=centro-1;
            else
                izquierda=centro+1;
            cout << " En la posicion izquierda" << izquierda; //Para seguimiento
            cout << " En la posicion derecha" << derecha; //Para seguimiento
        }
        system("PAUSE");
    }
    if(encontrado) return centro; else return -1;
    //Podriamos devolver bool true si lo encuentra false en caso contrario
}
```

```
En la posicion 5 esta el valor 6 En la posicion izquierda0 En la posicion derecha4Presione una tecla para continuar . .
.
En la posicion 2 esta el valor 3 En la posicion izquierda3 En la posicion derecha4Presione una tecla para continuar . .
.
En la posicion 3 esta el valor 4 En la posicion izquierda4 En la posicion derecha4Presione una tecla para continuar . .
.
En la posicion 4 esta el valor 5Presione una tecla para continuar . . .
Encontrado en posicion 4
-----
Process exited after 2.238 seconds with return value 0
Presione una tecla para continuar . . .
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

- - -

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

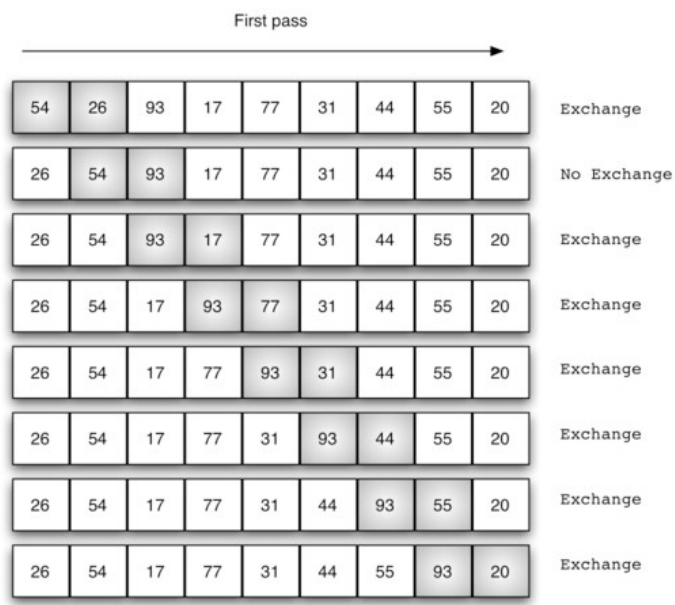


Ordenación

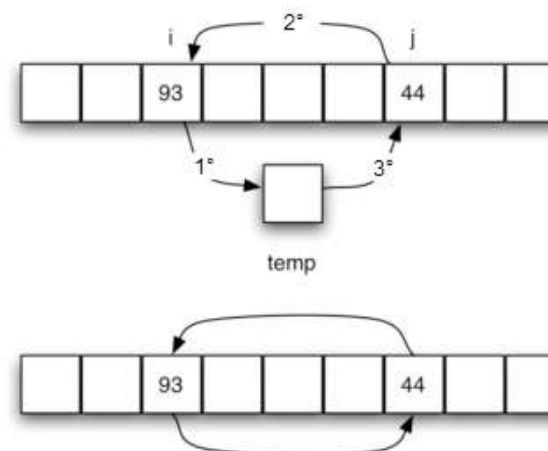
Burbuja.

Compara el elemento de una posición con el siguiente, si no están en orden los coloca.

Cada ejecución coloca el mayor al final del array. Hay que hacer $n-1$ comparaciones.



La mayoría de lenguajes de programación requieren un proceso de 3 pasos con una ubicación de almacenamiento extra.



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP: 689 45 44 70

Cartagena99