

Arreglos

Ordenamiento y Búsquedas

Fundamentos de programación

Búsqueda en un Arreglo

- Se refiere al proceso para encontrar un elemento particular en un arreglo.
- Una de las estrategias mas comunes y simples para buscar un dato en un arreglo es:
 - Revisar uno por uno los elementos del mismo, este método se conoce como **búsqueda lineal**.
- Escribir una función que determine si un valor dado se encuentra en un arreglo de elementos enteros, y si es así, indique su posición-

Búsqueda lineal

Dato a buscar: 58

A(1)	19
A(2)	12
A(3)	1
A(4)	2
A(5)	58
A(6)	100
A(7)	3
A(8)	4
A(9)	45
A(10)	25

```
encontrado=0
A=[19;12;1;2;58;100;3;4;45;25]
valor=input("Valor a buscar: ")
for i=1:10
    if (valor==A(i)) then
        disp(i)
        encontrado=1
    end
end
if (encontrado==0) then
    mprintf("El valor no se encuentra en el vector")
end
```

Búsqueda en un Arreglo

- La búsqueda lineal es simple pero tiende a ser muy lento ya que se requiere buscar en todo el arreglo, lo que hace el proceso muy largo.
- La **búsqueda binaria** es un método, donde:
 - La lista debe estar ordenada en un orden específico de acuerdo al valor de la llave.
 - Debe conocerse el número de registros.

Búsqueda Binaria

7	16	17	26	30	38	41	45	49	59	67	72	80	87	96	102	107	115	122
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

- Pasos

1. Se compara la llave buscada con la llave localizada al centro del arreglo.
2. Si la llave analizada corresponde a la buscada fin de búsqueda si no.
3. Si la llave buscada es menor que la analizada repetir proceso en mitad superior, sino en la mitad inferior.
4. El proceso de partir por la mitad el arreglo se repite hasta encontrar el registro o hasta que el tamaño de la lista restante sea cero , lo cual implica que el valor de la llave buscada no esta en la lista.

Búsqueda Binaria

Ejemplo de Búsquedas

- Se tiene un arreglo ordenado de 19 casillas.

7	16	17	26	30	38	41	45	49	59	67	72	80	87	96	102	107	115	122
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Si buscamos el número 107. ¿En que posición del arreglo se encuentra? ¿Cuántas comparaciones se hacen?

Búsqueda Secuencial

Posición = 17

Comparaciones = 17

Búsqueda Binaria

Posición = 17

comparaciones = 3

Búsqueda lineal vs. binaria

Búsqueda Lineal	Búsqueda Binaria
Sirve para realizar búsquedas en arreglos ordenados y no ordenados.	Sirve sólo para realizar búsquedas en arreglos ordenados.
Lento.	Rápido.
Bueno para arreglos pequeños.	Bueno para arreglos muy grandes.
Máximo n pasos, donde n es el número de elementos 30 elementos 30 pasos	Máximo k pasos donde $2^k > \#$ elementos 30 elementos 5 pasos

Búsqueda Binaria

```
function pos = busquedaBinaria(arreglo, clave, linf, lsup)
    pos=-1
    enc=0
    while (linf<=lsup & enc==0)
        mitad=linf+int((lsup-linf)/2)
        if (arreglo(mitad)==clave)
            enc=1
        else
            if (clave<arreglo[mitad])
                lsup=mitad-1
            else
                linf=mitad+1
            end
        end
    end
    if (enc==1)
        pos=mitad
    end
endfunction
```


Ordenamiento en Arreglos

- Ordenación tipo Burbuja
 - (ordenación por hundimiento) los valores mas pequeños “flotan” hacia la parte superior del arreglo.
 - Fácil de programar, lento en ejecución.
 - Técnica que consiste en llevar a cabo varias pasadas a través del arreglo. En cada pasada se comparan pares sucesivos de elementos.

Ejemplo: $a = [8, 12, 4, 9, 3]$

8	12	4	9	3
---	----	---	---	---

8	4	9	3	12
---	---	---	---	----

4	3	8	9	12
---	---	---	---	----

3	4	8	9	12
---	---	---	---	----

3	4	8	9	12
---	---	---	---	----

Ordenación tipo Burbuja

```
function []= ordenar_burbuja(arreglo, nroElementos)
for i = 1: nroElementos - 1
    for j = 2: nroElementos
        if (arreglo(j) < arreglo(j-1)) then
            aux_elem = arreglo(j)
            arreglo(j) = arreglo(j-1)
            arreglo(j-1)= aux_elem;
        end
    end
end
endfunction
```

Ordenamiento por selección

- Busca o **selecciona** el menor entre todos y lo intercambia con el primer elemento del arreglo.
- Así sucesivamente hasta quedarse con los dos últimos elementos del arreglo.

Ejemplo: $a = [8, 12, 4, 9, 3]$

8	12	4	9	3
---	----	---	---	---

3	12	4	9	8
---	----	---	---	---

3	4	12	9	8
---	---	----	---	---

3	4	8	9	12
---	---	---	---	----

3	4	8	9	12
---	---	---	---	----

Ordenación por Selección

```
function [arregloSalida]=selectionSort(arregloEntrada)
    //Copiamos el arreglo de entrada
    arregloSalida=arregloEntrada;
    //Para todos los elementos menos 1
    for i=1:(length(arregloSalida)-1)
        //Buscamos el menor en el rango actual
        [menor m]=obtenMenor(arregloSalida,i,length(arregloSalida));
        //Lo intercambio con el primero en el rango
        temp=arregloSalida(i);
        arregloSalida(i)=arregloSalida(m);
        arregloSalida(m)=temp;
    end
endfunction
```

```
//Funcion que retorna el elemento con el
menor valor dentro de un arreglo
function [menor,j]=obtenMenor(arreglo,ini,fin)
    menor=arreglo(ini)
    j=ini;
    for i=ini:fin
        if(arreglo(i)<menor) then
            menor=arreglo(i);
            j=i;
        end
    end
end
endfunction
```

En esta clase Ud. aprendió

- Resolver problemas asociados a la ordenación de arreglos.
- Identificar dos de los métodos más utilizados de ordenación de arreglo.
- Identificar dos de los métodos más utilizados en búsqueda de elementos en un arreglo.
- Identificar las ventajas y desventajas de cada uno de los métodos de búsqueda y ordenación de arreglos.