Taller de Programación

AGENDA

Método de ordenación: selección

ARREGLOS – Ordenación - SELECCION

Este algortimo consta de N vueltas, donde N es la dimension lógica del arreglo.

En la **primera Vuelta,** se recorre todo el arreglo desde la posición 1 hasta el final (dimL) y se guarda en que posición se encuentra el elemento mas chico del arreglo. Al terminar el recorrido se intercambia el elemento de la posición 1 con el elemento ubicado en la posición en la cual se encontró el menor valor.

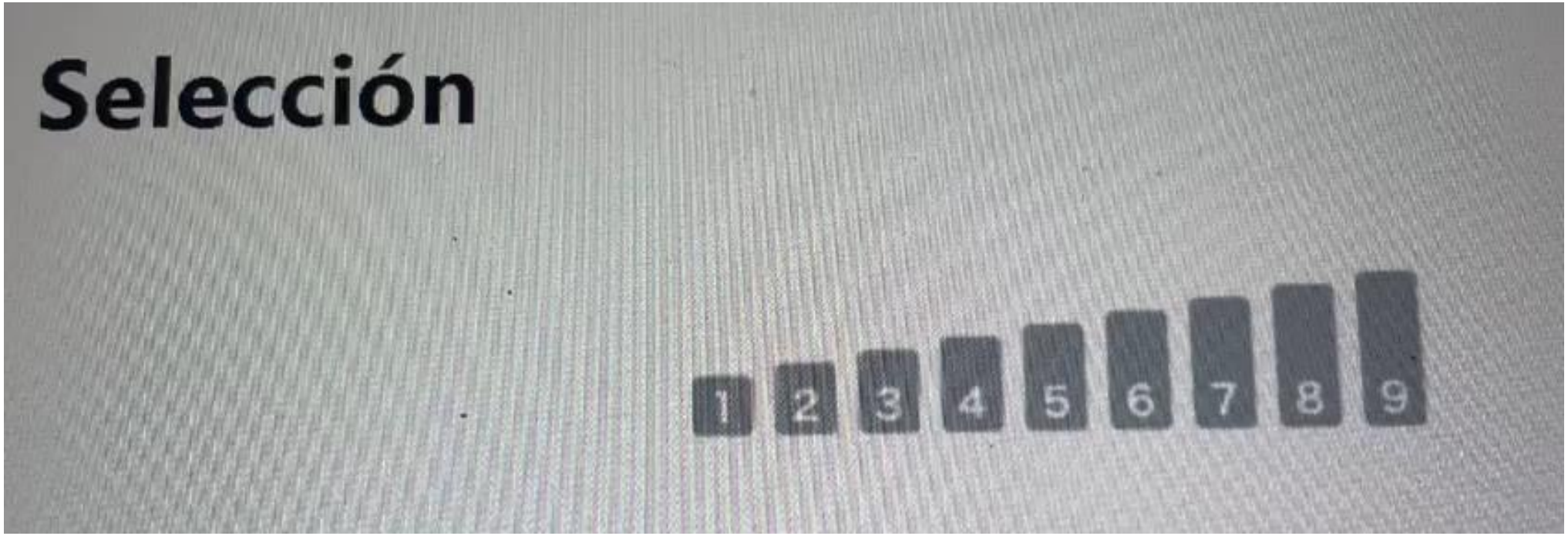
En la **segunda vuelta (ya se sabe, que en la posición 1 quedó ubicado el menor)**, y por lo tanto se busca a partir de la posición 2 hasta el final y se guarda en que posición se encuentra el elemento mas chico del arreglo. Al terminar el recorrido, se intercambia el elemento de la posición 2 con el elemento ubicado en la posición en la cual se encontró el menor valor.

Esto se repite hasta recorrer todo el arreglo, dando así un total de N vueltas (N= dimension logica del arreglo -1), ya que el último elemento en .la última vuelta ya está ubicado en su posición.

ARREGLOS – Ordenación - SELECCION

QUE NECESITAMOS CONOCER?

- Dimensión lógica del arreglo.
- Posición donde va el elemento mínimo.
- Rango en donde se busca el mínimo desde la que vamos a buscar el mínimo.
- Posición del elemento mínimo.

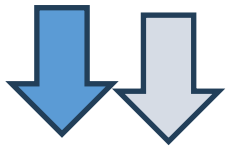


ARREGLOS – Ordenación - SELECCION

23	1	100	4	7	

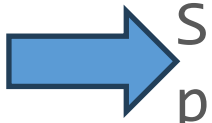
dimF = 6
dimL= 5

VUELTA 1



23	1	100	4	7	

mínimo = 1
pos= 2



Se intercambia el elemento de la posición 1 (23) con el de la posición 2 (1).

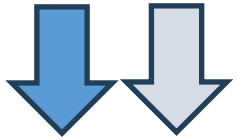
1	23	100	4	7	

ARREGLOS – Ordenación - SELECCION

1	23	100	4	7	

dimF = 6
dimL= 5

VUELTA 2



1	23	100	4	7	

mínimo = 4
pos= 4

Se intercambia el elemento de la posición 2 (23) con el de la posición 4 (4).

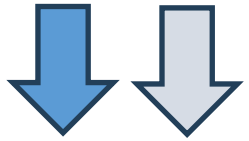
1	4	100	23	7	

ARREGLOS – Ordenación - SELECCION

1	4	100	23	7	

dimF = 6
dimL= 5

VUELTA 3



1	4	100	23	7	

mínimo = 7
pos= 5

Se intercambia el elemento de la posición 3(100) con el de la posición 5 (7).

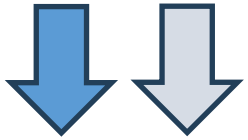
1	4	7	23	100	

ARREGLOS – Ordenación - SELECCION

1	4	7	23	100	

dimF = 6
dimL= 5

VUELTA 4



1	4	7	23	100	

mínimo = 23
pos= 4

Se intercambia el elemento de la posición 4(23) con el de la posición 4 (23).

1	4	7	23	100	

ARREGLOS – Ordenación - SELECCION

```
Program ordenar;

Const  dimF =... {máxima longitud del arreglo}

Type
      TipoElem = ... { tipo de datos del vector }
      Indice   =  0.. dimF;
      Tvector =  array  [ 1..dimF] of TipoElem;

Var
  a:Tvector;
  dimL:integer;

Begin
  cargarVector (a, dimL);
  seleccion (a, dimL);
End.
```

ARREGLOS – Ordenación - SELECCION

```
Procedure seleccion ( var v: tVector; dimLog: indice );

var i, j, pos: indice; item : tipoElem;

begin
 for i:=1 to dimLog-1 do begin {busca el mínimo y guarda en pos la posición}
           pos := i;
           for j := i+1 to dimLog do
              if v[ j ] < v[ pos ] then pos:=j;

           {intercambia v[i] y v[p]}
           item := v[ pos ];
           v[ pos ] := v[ i ];
           v[ i ] := item;
       end;
end;
```

ARREGLOS – Ordenación - SELECCION

```
Program ordenar;

Const  dimF = 200

Type
   vectorEnteros =  array  [ 1..dimF] of integer;

Var
  vE:vectorEnteros;
  dimL:integer;

Begin
  cargarVector (vE, dimL);
  seleccion (vE, dimL);
End.
```

ARREGLOS – Ordenación - SELECCION

```
Procedure seleccion ( var v: vectorEnteros; dimLog: integer);

var i, j, pos: integar; item : integer;

begin
 for i:=1 to (dimLog-1) do begin {busca el mínimo y guarda en pos la posición}
          pos := i;
          for j := i+1 to dimLog do
             if v[ j ] < v[ pos ] then pos:=j;

          {intercambia v[i] y v[p]}
          item := v[ pos ];
          v[ pos ] := v[ i ];
          v[ i ] := item;
      end;
end;
```

ARREGLOS – Ordenación - SELECCION

CONSIDERACIONES

- Tiempo de ejecución.
- Facilidad para la escritura del mismo.
- Memoria utilizada en su ejecución.
- Complejidad de las estructuras auxiliares que necesite.
- Requiere el mismo tiempo si los datos ya están ordenados, si están al azar, si se encuentran en el orden exactamente inverso al que yo los quiero tener.

- N^2.
- Muy fácil.
- El arreglo y variables.
- No requiere
- Siempre require el mismo tiempo de ejecución.