

Unidad 6

ESTRUCTURAS
ESTÁTICAS
ARRAYS

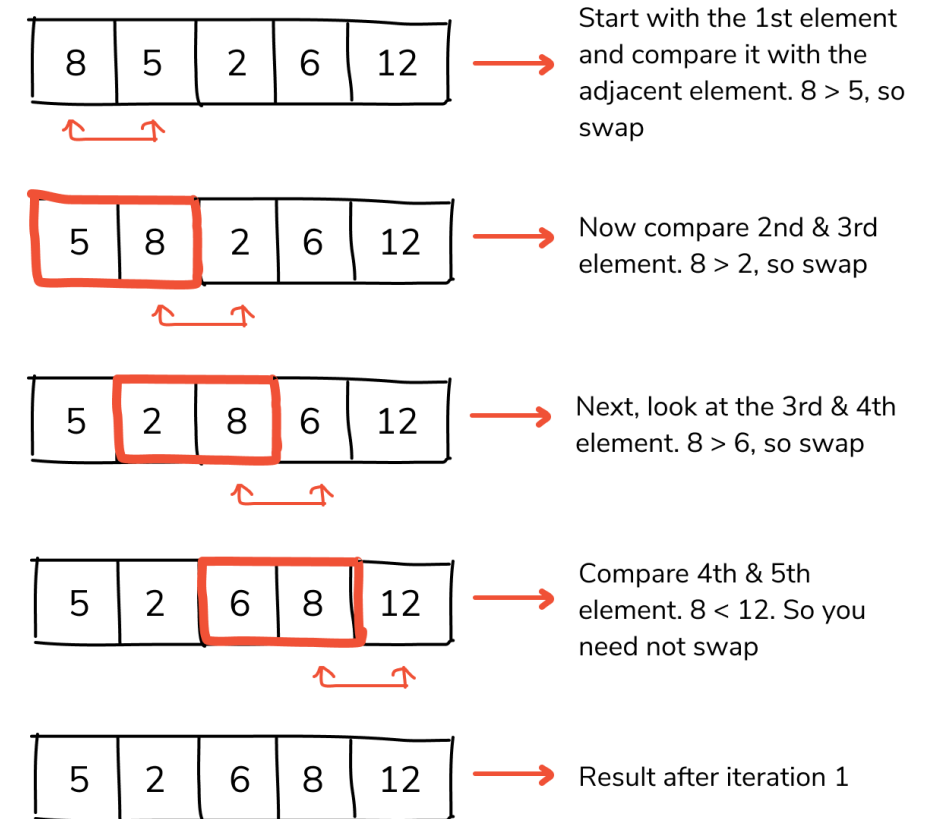
Arrays

ORDENAMIENTO Y BÚSQUEDA

Estructuras estáticas - Arrays

Ordenamiento por burbujeo

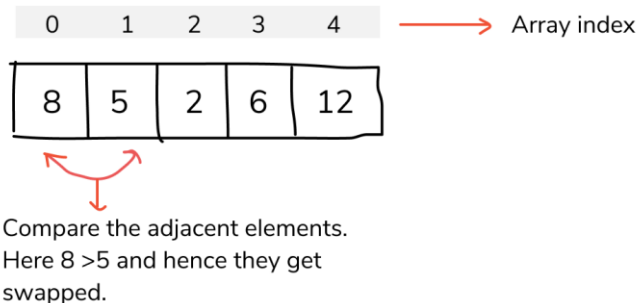
- Método de ordenamiento donde cada elemento del array será comparado con todo el array e intercambiando su posición, si es mayor que el elemento comparado.
- Luego de la primera iteración, habrá que seguir repitiendo este conjunto de acciones una y otra vez hasta que se ordene toda la matriz de elementos.
- En general tarda $(n-1)$ iteraciones en ordenar todo la matriz.



Estructuras estáticas - Arrays

Ordenamiento por burbujeo

- Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado.
- Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada



```
int miArray[]={8,5,2,6,12};
int i,y,aux=0;
int n = 5;
for(i=0;i<n-1;i++){ //recorro el array hasta dos posiciones antes del final
    for(y=0;y<n-i-1;y++){ //por cada posicion, vuelvo a recorrerlo todo

        //comparo con la posición siguiente, si es mayor...
        if (miArray[y]>miArray[y+1]){
            //utilizo una variable auxiliar para guardar temporalmente el valor actual
            aux=miArray[y];
            //guardo en la posicion actual, el valor de la siguiente
            miArray[y]=miArray[y+1];
            //guardo en la posicion siguiente el valor que teniamos en aux
            miArray[y+1]=aux;
        }
    }
}
```

```
for(i=0;i<n;i++){
    printf("El valor de la pos %d es %d\n",i,miArray[i]);
}
```



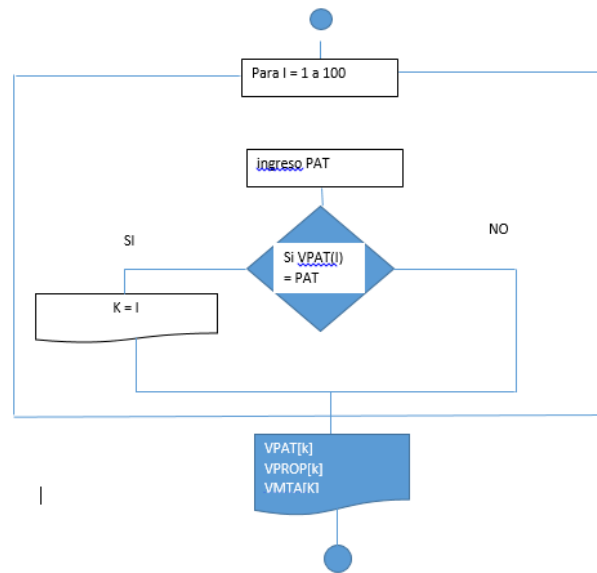
```
D:\Dropbox\UAI\grado\Programac
El valor de la pos 0 es 2
El valor de la pos 1 es 5
El valor de la pos 2 es 6
El valor de la pos 3 es 8
El valor de la pos 4 es 12

-----
Process exited after 0.06838 s
Presione una tecla para contin
```

Estructuras estáticas - Arrays

Búsqueda Secuencial

- Para buscar dentro de un vector **necesitamos conocer el índice**
- Utiliza un ciclo para **recorrer todo el vector hasta encontrar el elemento buscado** o bien el final.
- Se usa cuando el **vector no está ordenado** o bien no puede ordenarse.
- No se puede asegurar la existencia hasta recorrer todo el vector.



```
int busquedaSecuencial(int array[], int dato){
    int i;
    for(i=0;i<10;i++){

        if (array[i]==dato){

            return i;
        }
    }

    return -1;
}
```

Estructuras estáticas - Arrays

Búsqueda Binaria

- Se usa cuando el vector está ordenado.
- Optimiza el tiempo de búsqueda.
- Se compara con el elemento central del vector.
 - Si el elemento a buscar es mayor al central, se toma solo la mitad superior.
 - Si el elemento a buscar es menor al central, se toma solo el bloque inferior
- Se vuelve a analizar en el medio del bloque anteriormente indicado y se repite el proceso hasta que no se pueda dividir más

```
int busquedaBinaria(int vector[], int n, int dato) {  
    int centro, inf=0, sup=n-1;  
    while(inf<=sup){  
        centro=((sup-inf)/2)+inf;  
        if(vector[centro]==dato) return centro;  
        else if(dato < vector[centro]) sup=centro-1;  
        else inf=centro+1;  
    }  
    return -1;  
}
```

Ejemplo

ORDENAMIENTO POR BURBUJEO, BÚSQUEDA SECUENCIAL
Y BÚSQUEDA BINARIA UTILIZANDO FUNCIONES.

Gracias!