



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Tista Garcia Edgar

Profesor:

Estructura de Datos y Algoritmos II

Asignatura:

5

Grupo:

1

No de Práctica(s):

Calzada Martinez Jonathan Omar

Integrante(s):

*No. de Equipo de
cómputo empleado*

2019-2

Semestre:

Fecha de entrega:

Obervaciones:

CALIFICACIÓN: _____

En la librería de ordenamiento se encuentran las funciones de ordenamientos quickSort y bubbleSort, en ellas se encuentra el algoritmo correspondiente para ser llamado en la función principal según se requiera o se necesite.

```
void bubbleSort(int a[], int size){
    int intercambios=0;
    int i,j,n;
    n= size;
    for(i=n-1;i>0;i--){
        int cambios=0;
        for(j=0; j<i; j++){
            if(a[j]>a[j+1]){
                swap(&a[j], &a[j+1]);
                cambios = 1;
            }
            printArray(a,size);
        }
        if(cambios==0)
            break;

        printArray(a,size);
        printf("Fin Iteracion");
    }
}
```

También en la librería de ordenamiento encontramos una función llamada particion donde agarra el valor que se va a utilizar de pivote para empezar a hacer el ordenamiento.

Librería de UTILIDADES

En la librería de utilidades encontramos funciones que utilizan apuntadores para poder hacer los cambios de información en las variables a la hora de hacer los intercambios en los ordenamientos.

Así también imprime los resultados obtenidos de los ordenamientos ya sea utilizando el algoritmo de Burbuja o quickSort

```
#include <stdio.h>
void swap(int* a, int* b){
    int t = *a;
    *a = *b;
    *b = t;
}

void printArray(int arr[],int size){
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
```

En la funcion principal encontramos un arreglo en donde lo inicializamos con valores arbitrarios dados por el profesor.

```
int main () {
    int arr[] = {30,60,23,1,56,77,21,11,78,40};
    int size = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr,size);
    //quickSort(arr,0,size-1);
    printArray(arr,size);

    return 0;
```

Utilizando el codigo con la funcion Burbuja obtenermos estos resultados

```
30 60 23 1 56 77 21 11 78 40
30 23 60 1 56 77 21 11 78 40
30 23 1 60 56 77 21 11 78 40
30 23 1 56 60 77 21 11 78 40
30 23 1 56 60 77 21 11 78 40
30 23 1 56 60 21 77 11 78 40
30 23 1 56 60 21 11 77 78 40
30 23 1 56 60 21 11 77 78 40
30 23 1 56 60 21 11 77 40 78
30 23 1 56 60 21 11 77 40 78
Fin Iteracion23 30 1 56 60 21 11 77 40 78
23 1 30 56 60 21 11 77 40 78
23 1 30 56 60 21 11 77 40 78
23 1 30 56 60 21 11 77 40 78
23 1 30 56 21 60 11 77 40 78
23 1 30 56 21 11 60 77 40 78
23 1 30 56 21 11 60 77 40 78
23 1 30 56 21 11 60 40 77 78
```

Utilizando la funcion quickSort obtenermos estos resultados

```
1 //
2 // main.c
3 // prsactica1
4 //
5 // Created by Alumno on 2/5/19.
6 // Copyright © 2019 alumno. All rights reserved.
7 //
8
9 #include <stdio.h>
10 #include "ordenamientos.h"
11
12
13 int main () {
14     int arr[] = {30,60,23,1,56,77,21,11,78,40};
15     int size = sizeof(arr)/sizeof(arr[0]);
16     //bubbleSort(arr,size);

```

Program ended with exit code: 0Pivote:40
Sub array : 30 23 1 21 11
Sub array : 60 56 78 77
Pivote:11
Sub array : 1
Sub array : 30 21 23
Pivote:23
Sub array : 21
Sub array : 30
Pivote:77
Sub array : 60 56
Sub array : 78
Pivote:56
Sub array :
Sub array : 60
1 11 21 23 30 40 56 60 77 78
Program ended with exit code: 0