

INFORME:

```
mer.c > ...
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define maxN 100000+3 /* Maximo tamaño de arreglo de entrada. */

int a[maxN],tmp[maxN],acpy[maxN]; /* El arreglo "a" guarda el arreglo de la entrada.txt, "tmp" es el arreglo auxiliar
para el merge sort, "acpy" es una copia del arreglo "a" para cuando se ordena volverlo a su estado original y volverlo
a ordenar con otro algoritmo distinto. */

unsigned long long int ifCnt,swapCnt; /* "ifCnt" cuenta cuantas veces se hace un condicional, "swapCnt" cuantas veces
se hace un intercambio de datos. */
```

- Incluí la librería time.h para el reporte de tiempos, no recuerdo la verdad si la vimos en clase, pero fue curioso porque yo sigo muchas páginas de informática en internet donde explican las cosas y en una de ellas mencionaban que el time.h formateaba el tiempo y hora del sistema por lo que decidí utilizarla, en el reporte de tiempos aparece en minutos, al principio me pareció raro que me apareciera en 0.0 pero luego de varios intentos me di cuenta que era lo que demoraba, porque hice la comparación de mi tiempo con el de la computadora y eran semejantes.
- Se me complicó la parte del manejo del tiempo con la librería porque es algo nuevo y tuve que leer varias paginas de internet y ver algunos vídeos para poder manejarlo, de igual forma comenté todo para que fuera más amena la lectura del código.

```
void bubble_sort2 ( int N ){ /*bubble sort mejorado*/
    int i, j, tmp2, swap;
    for( i = 0; i < N-1;i++ ){ /*cuando el arreglo esta ordenado o parcialmente ordenado
        no se hacen todas las veces el ciclo de adentro*/
        swap = 0;
        for( j = 0; j < N-i-1;j++ ){
            if( a[ j ] > a[ j+1 ] ){
                tmp2 = a[ j+1 ];
                a[ j+1 ] = a[ j ];
                a[ j ] = tmp2;
                swapCnt++;
                swap = 1;
            }
            ifCnt++;
        }
        if( swap == 0 ) break; /*si no se hizo ningun swap en el ciclo de adentro el arreglo esta ordenado*/
    }
}
```

- Las variables de swap las utilicé para mejorar el algoritmo de Bubblesort que era un punto que se pedía mejorar cualquier algoritmo, la verdad se me complicó mucho porque yo pensaba modificar algo más profundo que unas iteraciones, pero ya sería involucrarme en cambiar el modo del ordenamiento, dejando de ser Bubblesort, en pocas palabras lo que hice como ven ahí comentado fue evitar que hiciera ciclos de más cuando ya se encuentra ordenado o casi y en segunda instancia lo que hice fue que al no hacer el swap es que ya se encuentra ordenado, evitando nuevamente hacer ciclo innecesarios.

IMPORTANTE:

- Aprendí a manejar la librería time.h
- Entendí mejor los algoritmos de ordenamientos y cuales puedo utilizar en ciertas circunstancias, no irme por el primero que piense y sea fácil.
- Los algoritmos de ordenamiento cuadrados tienden a ser más demorados.
- Los algoritmos con datos pequeños no tardan, pero con grandes datos se nota la diferencia entre cada uno.
- El merge sort es el mejor para datos grandes.