



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
ING. EN SISTEMAS
COMPUTACIONALES

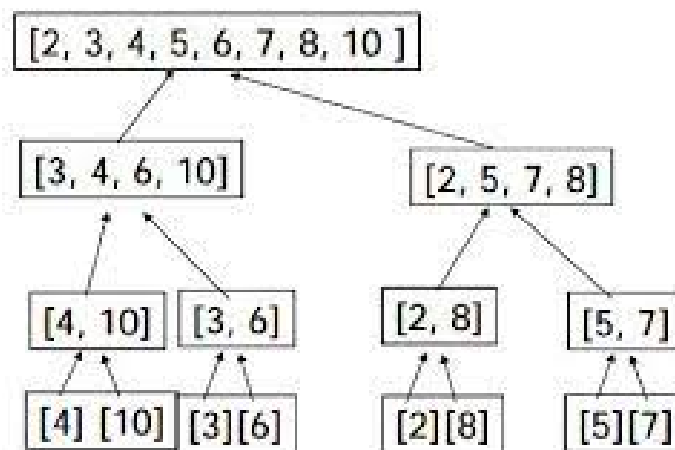


Práctica No. 2

ALGORITMOS DIVIDE Y VENCERÁS

ANÁLISIS Y DISEÑO DE ALGORITMOS
3CM3

PROFESORA: SÁNCHEZ GARCÍA LUZ MARÍA



Alumnos:

Bastida González María Guadalupe
Mejía Romo Pablo Emanuel

Boletas:

2021630164
2020350970

Práctica No. 2: ALGORITMOS DIVIDE Y VENCERÁS

Objetivo: Programar algoritmos recursivos con la estrategia Divide y vencerás para la solución de problemas computacionales.

Actividades:

1. Programar en ANSI C 3 algoritmos Divide y vencerás.
2. Utilizar funciones recursivas para la programación de los algoritmos, es decir, no se admitirán algoritmos iterativos.
3. Realizar un análisis temporal para cada algoritmo, indicando la ecuación de recurrencia y la notación de orden (Big O).

Algoritmo MergeSort (Mezcla) y QuickSort (Rápido).

Planteamiento del problema: Se tienen listas de números desordenados dentro del vector, ambos algoritmos buscan dar orden a estas listas.

$$\text{MergeSort } T(n) \begin{cases} 1, & n=1 \\ 2T(n/2)+n, & n>1 \end{cases} \therefore O(n \log_2(n))$$

$$\begin{array}{ll} \text{QuickSort} & \text{Peor caso: } T(n) = T(1) + T(n-1) + c \cdot n \in O(n^2) \\ & \text{Caso medio: } T(n) = 2T(n/2) + c \cdot n \in O(n \log(n)) \end{array}$$

Algoritmo Búsqueda binaria (previo ordenamiento del arreglo).

Planteamiento del problema: se tiene un arreglo de números ordenado de manera ascendente, busca una estrategia para encontrar la posición de un número x dentro de un arreglo.

$$T(n) = 3T(n/2) + n + 1 \therefore O(n \log(n))$$

Algoritmo máximo y mínimo en un arreglo.

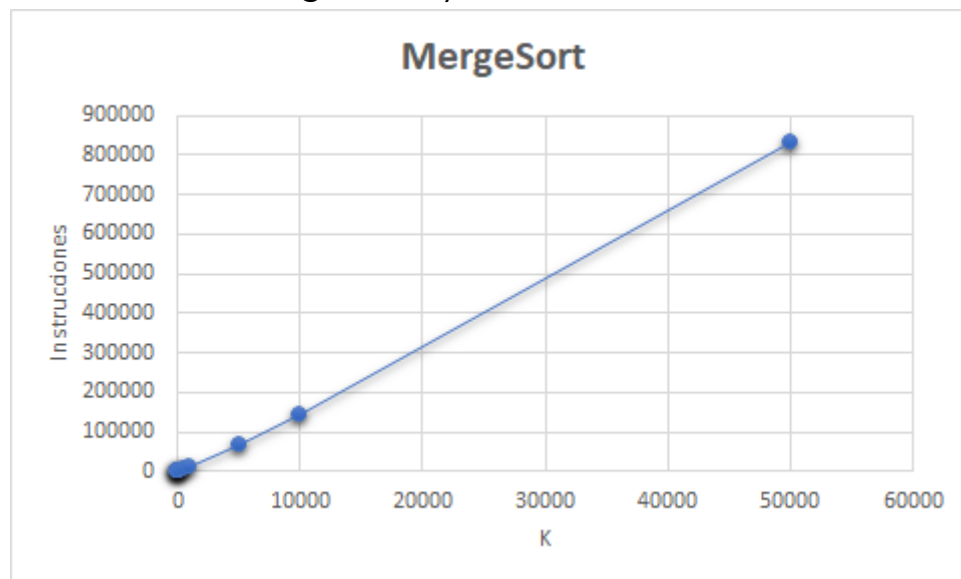
Planteamiento del problema: Se tiene un arreglo con valores desordenados, representando cualquier dato, se necesita encontrar el valor máximo y mínimo dentro del arreglo.

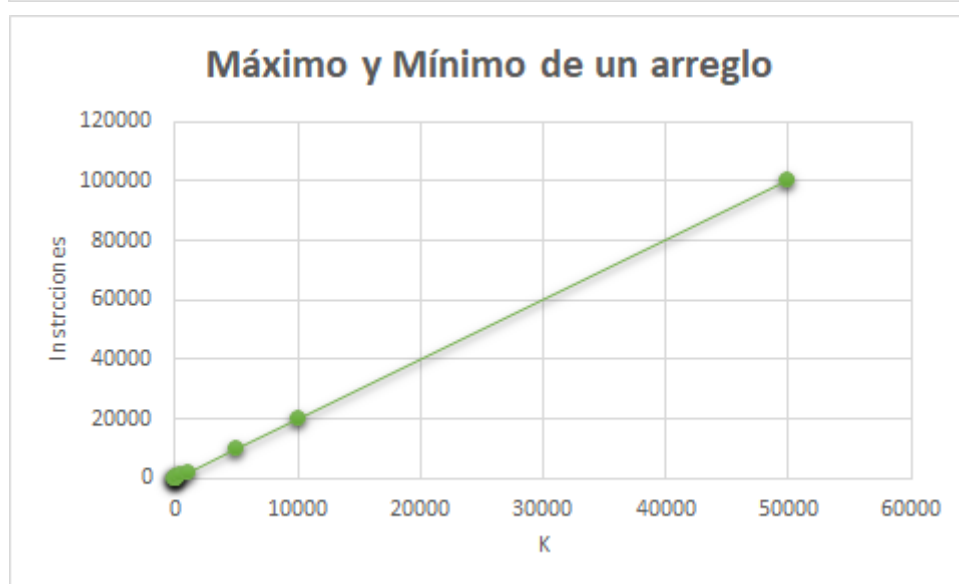
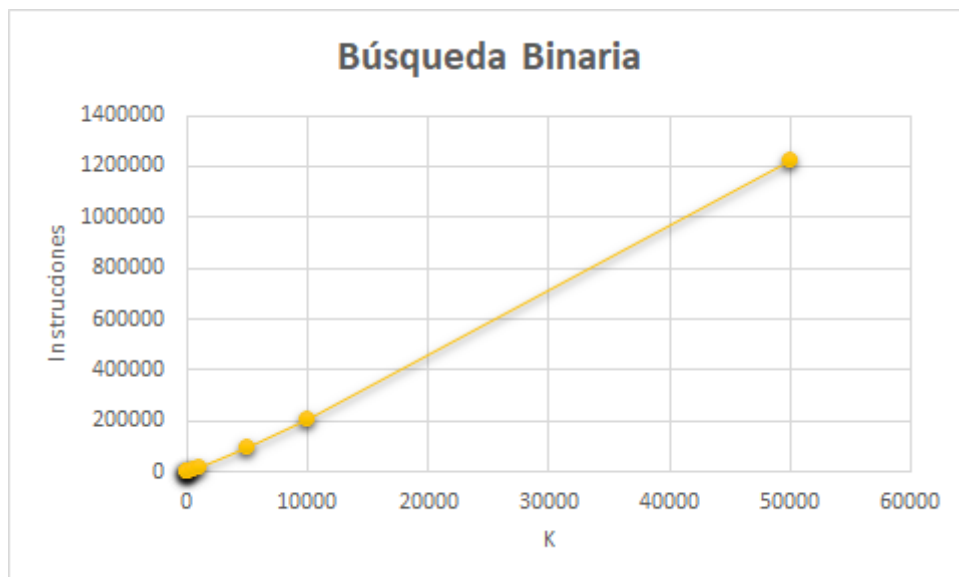
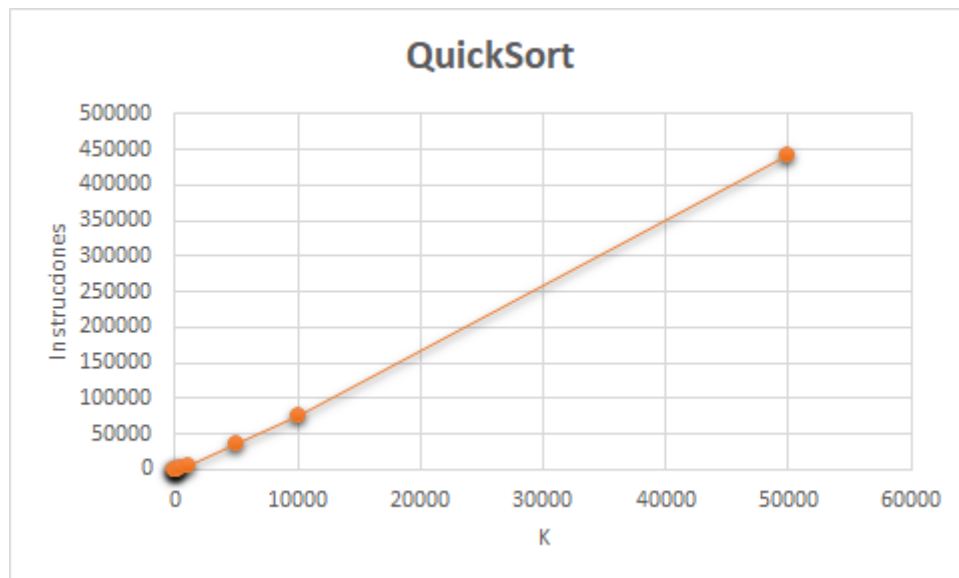
$$T(n) = 2T(n/2) + 2 \therefore O(n)$$

4. Calcular la cantidad de instrucciones que tarda cada algoritmo en realizar las operaciones.

	Instrucciones			
K	A.MergeSort	A.QuickSort	A.Búsqueda Binaria con QuickSort	A.Máx y min de un arreglo
10	43	27	71	18
20	106	63	164	38
30	177	104	267	58
40	253	146	376	78
100	764	432	1118	198
500	4283	2741	7252	998
1000	10966	5983	15980	1998
5000	66439	35719	97196	9998
10000	142877	76439	209357	19998
50000	830582	440241	1220771	99998

5. Graficar el comportamiento temporal de cada algoritmo (1 gráfica para cada uno de los algoritmos).





6. Realizar un análisis espacial para cada algoritmo en notación de orden Big O.

Algoritmo MergeSort (Mezcla) y QuickSort (Rápido).

$$E(n) = n+11 \therefore O(n)$$

$$E(n) = n+7 \therefore O(n)$$

Algoritmo Búsqueda binaria (previo ordenamiento del arreglo).

$$E(n) = n+12 \therefore O(n)$$

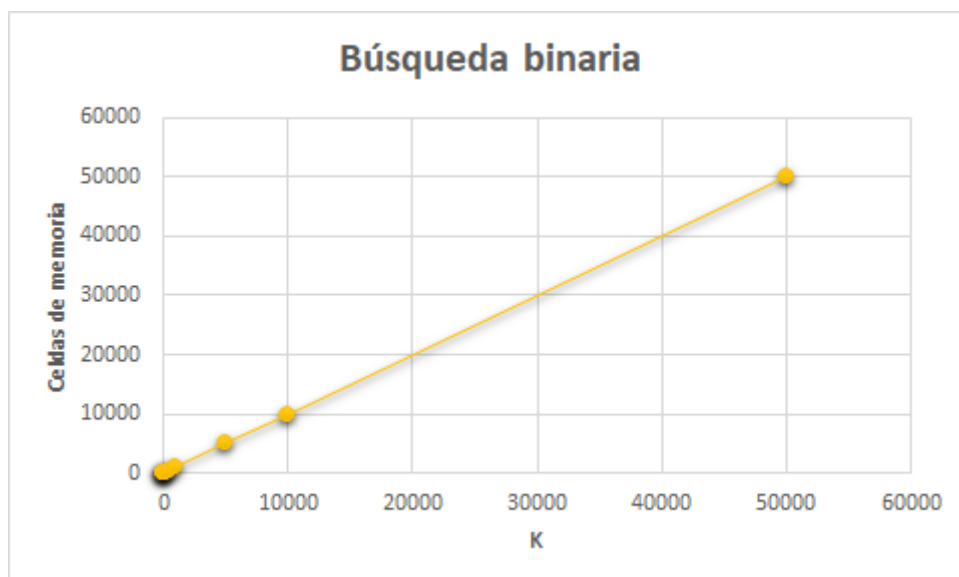
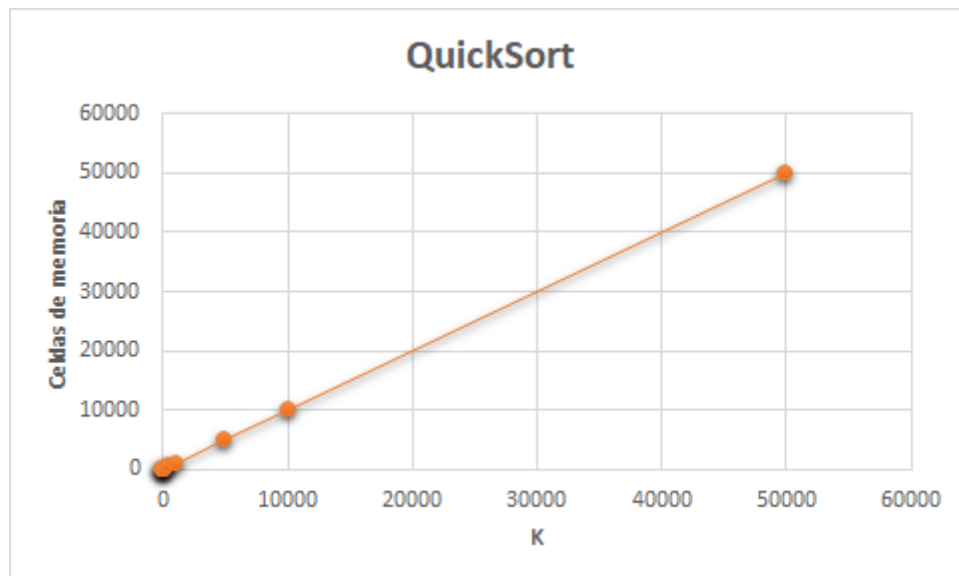
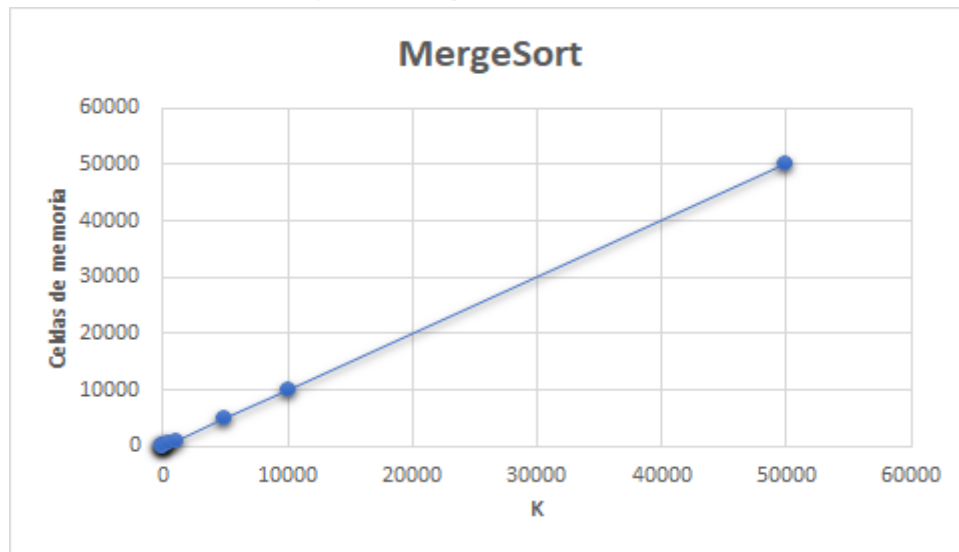
Algoritmo máximo y mínimo en un arreglo.

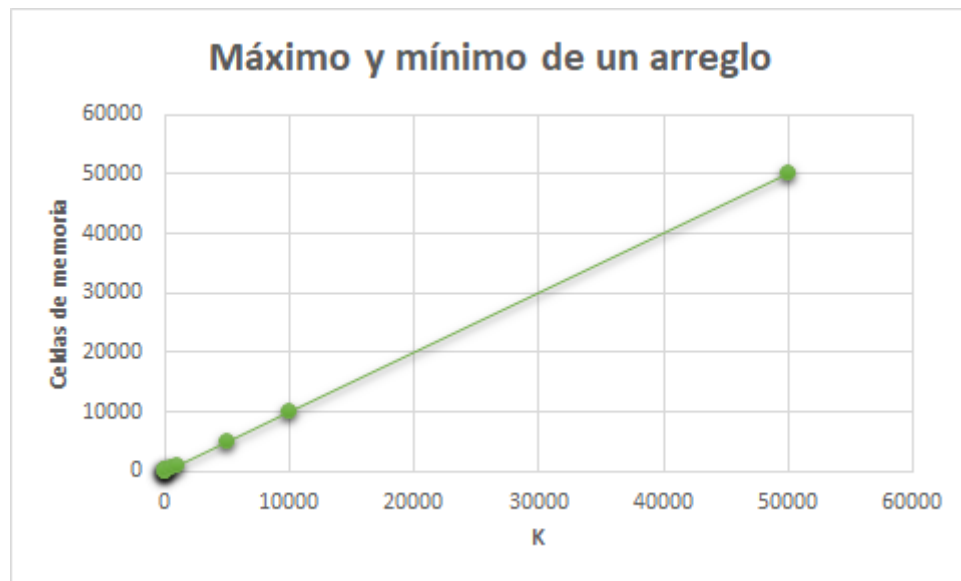
$$E(n) = n+4 \therefore O(n)$$

7. Calcular la cantidad de celdas de memoria necesarias para cada algoritmo.

	Celdas			
K	A.MergeSort	A.QuickSort	A.Búsqueda Binaria con QuickSort	A.Máy y min de un arreglo
10	21	17	22	14
20	31	27	32	24
30	41	37	42	34
40	51	47	52	44
100	11	107	112	104
500	511	507	512	504
1000	1011	1007	1012	1004
5000	5011	5007	5012	5004
10000	10011	10007	10012	10004
50000	50011	50007	50012	50004

8. Graficar el comportamiento espacial de cada algoritmo (1 gráfica para cada uno de los algoritmos).





9. Indique el entorno controlado para realizar las pruebas experimentales.

Se utilizó el editor de códigos Visual Studio Code junto con la terminal de comandos que esta incorpora (CMD) para la programación de los códigos en c, el equipo de cómputo donde se ejecutaron dichos códigos cuenta con las siguientes especificaciones:

- Procesador: Intel Core i7-4600 de 4 núcleos a 2.69 GHz
- Memoria RAM: DDR3 de 8 GB
- Unidad de almacenamiento: Unidad de estado sólido de 500 GB
- Sistema operativo: x64 o 64 bits
- Compilador: MinGw32-gcc v6.3.0-1
- Editor de códigos: Visual Studio Code v1.55.2

Además, se procuró que el equipo no ejecutara otros programas más que un editor de código, una consola de comandos y el mismo algoritmo.

10. Responda las siguientes cinco preguntas:

a) Para cada algoritmo ¿Existe alguna solución que no sea recursiva? ¿Cuál es?

Para los algoritmos MergeSort y QuickSort no, ya que emplean un paradigma algorítmico común que justamente, se basa en la recursividad. El algoritmo de búsqueda binaria por su parte sigue necesitando recursividad para resolverse, si bien funciona con arreglos ordenados su funcionamiento se basa en dividir el arreglo

repetitivamente, de esta forma es necesaria la recursividad. Ahora bien para el algoritmo de máximo y mínimo de un arreglo es posible utilizar un algoritmo no recursivo para su resolución, puede ser resuelto de forma iterativa, por ejemplo mediante el algoritmo de selección.

b) ¿Cuál de los 3 algoritmos es más fácil de implementar?

El algoritmo máximo y mínimo en un arreglo.

c) ¿Cuál de los 2 algoritmos es el más difícil de implementar?

El algoritmo de búsqueda binaria con previo ordenamiento del arreglo.

d) ¿El comportamiento experimental de los algoritmos era el esperado? ¿Por qué?

Si, ya teníamos conocimiento del comportamiento en su mayoría logarítmico que tenían estos algoritmos, esto debido a que antes vimos algoritmos de organización. Así mismo, conocíamos su complejidad espacial y temporal al programarlos.

e) ¿Qué recomendaciones darían a nuevos equipos para realizar esta práctica?

Tener en cuenta la estrategia que usa y su algoritmo, es este caso abordar los problemas con divide y vencerás sin saber el logaritmo base dificultará las actividades, así mismo entender primero qué es lo que quiere resolver en el problema planteado y por último saber programar con recursividad.

Pruebas

● MergeSort

```
SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  PROBLEMAS
1: cmd

Microsoft Windows [Versión 10.0.19042.928]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc mergeSort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
3 4 5 6 9 9 10 10 10
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc mergeSort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
3 3 3 6 7 8 8 8 10 12 13 13 14 15 15 17 17 17 17 20
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc mergeSort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
2 2 4 6 9 9 12 13 13 14 14 16 17 18 18 19 20 20 20 22 23 24 2
4 25 25 27 27 30 30
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc mergeSort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
2 3 4 6 6 7 10 10 11 11 12 12 13 13 13 14 15 15 17 18 19 22 23 2
3 24 25 27 29 30 30 30 32 32 32 33 34 34 34 39 40
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>
```

● QuickSort

```
SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  PROBLEMAS
1: cmd

Microsoft Windows [Versión 10.0.19042.928]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc quicksort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
1 2 2 3 3 5 6 7 8
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc quicksort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
1 2 2 3 4 5 5 6 7 9 9 14 15 18 18 18 19 19 20 20
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc quicksort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
1 2 4 5 7 9 10 10 10 12 13 17 17 18 18 19 19 20 20 21 21 23 23 2
6 27 27 28 29 30 30
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc quicksort.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
1 3 4 4 5 5 6 7 7 8 8 13 13 13 13 13 13 15 17 18 18 19 20 2
1 22 22 24 25 25 27 28 28 32 33 34 34 37 37 37 38
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>
```

● Búsqueda binaria

```
SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  PROBLEMAS
1: cmd

(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc busqueda_binaria.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
2 2 2 3 4 6 6 7 8 10
El numero 4 se encuentra en la posicion 4 del arreglo

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc busqueda_binaria.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
1 3 3 3 4 5 7 8 8 9 10 10 11 12 13 14 15 15 18 19
El numero 8 se encuentra en la posicion 7 del arreglo

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc busqueda_binaria.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
2 2 2 3 4 5 6 7 7 7 10 11 12 12 13 14 15 16 16 17 21 22 23 2
3 25 26 27 27 30 30
El numero 25 se encuentra en la posicion 24 del arreglo

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc busqueda_binaria.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
3 5 6 6 7 7 10 10 13 13 14 15 15 16 16 17 19 20 20 20 20 20 21 2
3 26 26 27 28 28 29 29 30 30 31 33 35 35 39 39 40
El numero 12 no se encuentra en el arreglo
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>
```

- **Máximo y mínimo de un arreglo**

```
SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  PROBLEMAS
Microsoft Windows [Versión 10.0.19042.928]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc max_min.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
4      5      3      9      8      2      8      1      3      1
El maximo del arreglo es: 9
El minimo del arreglo es: 1
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc max_min.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
7      2      11     12     16      2      3     15     16     20      3     14      6      5     13      3     16      1     13     10
El maximo del arreglo es: 20
El minimo del arreglo es: 1
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc max_min.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
3      26     12     11     28     28     13      5     26      4     10     16      7      4     22     16     12      2     18     17     23     19     22     2
5      9      2      11     30      6      9
El maximo del arreglo es: 30
El minimo del arreglo es: 2
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>gcc max_min.c

C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>a.exe
32     15      6     39     39     32     36     25      8     23     25     35     40     32      4     25      8      9     19     11      7     22     12      2
6      15     35     40     40     29      4     25     32     37      9     30     14     30     22     13     23
El maximo del arreglo es: 40
El minimo del arreglo es: 4
C:\Users\emanu\Documents\ESCOM\3er_semestre\Algoritmos\Divide_y_venceras>
```

Anexo.

Para la ejecución del algoritmo, solamente es necesario compilar el archivo .c, ya sea en un IDE o en cons