

Tarea 1 divide y vencerá
Fidel Morales Briones A01198630

Maximin.cpp

caso de prueba 1:

```
PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\Maximin.exe caso1_maximin.txt
13
-3
-25
20
-3
-16
-23
18
20
-7
12
-5
-22
15
-4
7
El m  nimo del arreglo es: -25 y el m  ximo es: 20
```

caso de prueba 2:

```
PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\Maximin.exe caso2_maximin.txt
1
-2
-2
2
-23
-13
-2
14
28
-70
122
-4
-2
125
-43
72
El m  nimo del arreglo es: -70 y el m  ximo es: 125
```

caso de prueba 3:

```
PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\Maximin.exe caso3_maximin.txt
-90
1
3
-45
120
34
-3
56
-80
30
El m  nimo del arreglo es: -90 y el m  ximo es: 120
```

caso de prueba 4:

```

PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\Maximin.exe caso4_maximin.txt
153
-334
-2
2023
-345
-1656
-233
188
202
-7
123
-53
-222
1512
-443
78
El m  nimo del arreglo es: -1656 y el m  ximo es: 2023

```

MaximumSubarray.cpp

caso de prueba 1:

```

PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\MaximumSubarray.exe caso1_maximum_subarray.txt
13
-3
-25
20
-3
-16
-23
18
20
-7
12
-5
-22
15
-4
7
La suma del m  ximo subarray es: 43

```

caso de prueba 2:

```

PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\MaximumSubarray.exe caso2_maximum_subarray.txt
13
-3
-25
-3
-16
18
20
-7
-5
-22
15
7
La suma del m  ximo subarray es: 38

```

caso de prueba 3:

```

PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\MaximumSubarray.exe caso3_maximum_subarray.txt
1
-2
-2
2
-23
-13
-2
14
28
-70
122
-4
-2
125
-43
72
La suma del máximo subarray es: 270

```

caso de prueba 4:

```

PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\MaximumSubarray.exe caso4_maximum_subarray.txt
-90
1
3
-45
120
34
-3
56
-80
30
La suma del máximo subarray es: 207

```

caso de prueba 5:

```

PS C:\Users\fidel\Desktop\tec uni\5to\algoritmos\tareas\tarea 1 divide y venceras> .\MaximumSubarray.exe caso5_maximum_subarray.txt
153
-334
-2
2023
-345
-1656
-233
188
202
-7
123
-53
-222
1512
-443
78
La suma del máximo subarray es: 2023

```

Análisis de complejidad en el tiempo:

Maximin.cpp

a: 2 maximin hace dos llamadas recursivas

b: 2 el arreglo se divide en dos en cada llamada recursiva

$f(n)$: $O(1)$

$$T(n) = 2T\left(\frac{n}{2}\right) + 1, n > 2$$

$$T(n) = 1, n \leq 2$$

Caso 1 teorema maestro:

revisar si, $1 = O(n^{\log_2 2 - \epsilon})$

$1 = O(n^{1-\epsilon})$, es verdadero para $\epsilon = 0.1$

Por lo tanto la complejidad es de $T(n) = \Theta(n^{\log_2 2}) = \Theta(n)$

MaximumSubarray.cpp

a: 2 maximum subarray hace dos llamadas recursivas

b: 2 el arreglo se divide en dos en cada llamada recursiva

f(n): n, find_max_crossing_subarray tiene una complejidad de n porque itera sobre todo el subarreglo generado

$$T(n) = 2T\left(\frac{n}{2}\right) + n, n > 1$$

$$T(n) = 1, n = 1$$

Caso 2 teorema maestro:

revisar si, $n = \Theta(n^{\log_2 2}) =$

$n = \Theta(n)$ es verdadero

Por lo tanto la complejidad es de $T(n) = \Theta(n^{\log_2 2} \log_2 n) = \Theta(n \log_2 n)$