Maximin (A)

if $|A| = 2$
    return (max (A[1], A[2]), min (A[1], A[2])) $O(1)$

else
    Dividir A en dos subarreglos A_1 y A_2
    iguales (cada uno con mitad de elementos)

    $(max1, min1) = Maximin(A_1)$ $O(n/2)$
    $(max2, min2) = Maximin(A_2)$ $O(n/2)$

    return (max(max1, max2), min(min1, min2) $O(1)$

$$T(n) = \begin{cases} 2T(n/2) + 1 & , n > 2 \\ 1 & , n = 2 \end{cases}$$

1. $f(n) = O(n^{\log_b a - \varepsilon})$, $\varepsilon > 0$

$$1 = O(n^{\lg 2 - \varepsilon}), \quad \varepsilon > 0$$
$$1 = O(n^{1-\varepsilon}) \checkmark, \quad \varepsilon = 0.1$$

$$T(n) = \Theta(n^{\log_2 2})$$

$$\boxed{\Theta(n)}$$

FMCS (A, low, mid, high)

```
left-sum = -∞                        O(1)
sum = 0                              O(1)
for i = mid down to low              O(n/2)
    sum += A[i]                      O(1)
    if sum > left-sum                O(1)
        left-sum = sum               O(1)
        max-left = i                 O(1)

right-sum = -∞                       O(1)
sum = 0                              O(1)
for j = mid+1 to high                O(n/2)
    sum += A[j]                      O(1)
    if sum > right-sum               O(1)
        right-sum = sum              O(1)
        max-right = j                O(1)

return (max-left, max-right, left-sum + right-sum)
```

$O(n/2) \approx \boxed{O(n)}$

FMS (A, low, high)

Find Max Subarray (A, low, high):

```
if high == low                               O(1)
    return (low, high, A[low])               O(1)
else
    mid = ⌊(low + high)/2⌋                   O(1)

  (ll, lh, ls) = FMS (A, low, mid)           O(n/2)
  (rl, rh, rs) = FMS (A, mid +1, high)       O(n/2)
  (cl, ch, cs) = FMCS (A, low, mid, high)    O(n)

  if ls ≥ rs and ls ≥ cs                     O(1)
       return (ll, lh, ls)                   O(1)
  elseif rs ≥ ls and rs ≥ cs                 O(1)
       return (rl, rh, rs)                   O(1)
  else                                       O(1)
       return (cl, ch, cs)                   O(1)
```

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

caso 2: $f(n) = \Theta(n^{\log_b a})$

$O(n) = \Theta(n^{\log_2 2})$ $\qquad \Theta(n^{\log_b a} \log n)$

$O(n) = \Theta(n^1)$

$$\boxed{\Theta(n \log_2 n)}$$