4.1 最大子数组问题

```
FIND-MAX-CROSSING-SUBARRAY($A, low, mid, high$)
 1  $left\text{-}sum = -\infty$
 2  $sum = 0$
 3  for $i = mid$ downto $low$
 4      $sum = sum + A[i]$
 5      if $sum > left\text{-}sum$
 6          $left\text{-}sum = sum$
 7          $max\text{-}left = i$
 8  $right\text{-}sum = -\infty$
 9  $sum = 0$
10  for $j = mid + 1$ to $high$
11      $sum = sum + A[j]$
12      if $sum > right\text{-}sum$
13          $right\text{-}sum = sum$
14          $max\text{-}right = j$
15  return $(max\text{-}left, max\text{-}right, left\text{-}sum + right\text{-}sum)$
```

```
FIND-MAXIMUM-SUBARRAY($A, low, high$)
 1  if $high == low$
 2      return $(low, high, A[low])$
 3  else $mid = \lfloor (low + high)/2 \rfloor$
 4      $(left\text{-}low, left\text{-}high, left\text{-}sum)$ = FIND-MAXIMUM-SUBARRAY($A, low, mid$)
 5      $(right\text{-}low, right\text{-}high, right\text{-}sum)$ = FIND-MAXUMUM-SUBARRAY($A, mid + 1, high$)
 6      $(cross\text{-}low, cross\text{-}high, cross\text{-}sum)$ = FIND-MAXIMUM-SUBARRAY($A, low, mid, high$)
 7      if $left\text{-}sum \geq right\text{-}sum$ and $left\text{-}sum \geq cross\text{-}sum$
 8          return $(left\text{-}low, left\text{-}high, left\text{-}sum)$
 9      elseif $right\text{-}sum \geq left\text{-}sum$ and $right\text{-}sum \geq cross\text{-}sum$
10          return $(right\text{-}low, right\text{-}high, right\text{-}sum)$
11      else return $(corss\text{-}low, cross\text{-}high, cross\text{-}sum)$
```

分治算法的分析

首先，第 1 行花费常量时间，对于 $n = 1$ 的基本情况，也很简单：第 2 行花费常量时间，因此，

$$T(1) = \Theta(1)$$