

# Homework 5

Lei Zhang

March 16, 2016

## 1 Exercises 5.1 - 1

**a**

```
//Input: An array A[0..n-1] of comparable elements
//Output: The number of the largest element in the array
function MAX(A[0..n - 1])
    if n == 1 then
        return 0
    else if A[Max(A[0..⌊n/2 - 1⌋])] > A[Max(A[⌊n/2⌋..n])] then
        return Max(A[0..⌊n/2 - 1⌋])
    else
        return Max(A[⌊n/2⌋..n])
    end if
end function
```

**b**

The output will be the leftmost largest element.

**c**

$$C(n) = 2C(\lfloor n/2 \rfloor) + 1 \text{ for } n > 1, C(1) = 0$$

It is easy to find the exact solution to the worst-case recurrence for  $n = 2^k$ :

$$C_{worst}(n) = n \log_2 n - n + 1$$

**d**

The divide-and-conquer algorithm and brute-force algorithm have the same time complexity. They also have exactly the same number of elementary operations.

## 2 Exercises 5.1 - 2

a

```
//Input: An array A[0..n-1] of comparable elements
//Output: The smallest and largest elements in the array, [min, max]
function MINANDMAX(A[0..n - 1])
    if n == 1 then
        return [A[0], A[0]]
    else if n == 2 & A[0] <= A[1] then
        return [A[0], A[1]]
    else if n == 2 & A[0] > A[1] then
        return [A[1], A[0]]
    else
        min, max = MinAndMax(A[0..⌊n/2 - 1⌋])
        min2, max2 = MinAndMax(A[⌊n/2⌋..n])
        if min2 < min then
            min ← min2
        end if
        if max2 < max then
            max ← max2
        end if
        return [min, max]
    end if
end function
```

b

$$C(n) = 2C(n/2) + 2 \text{ for } n > 2, C(2) = 1, C(1) = 0$$

It is easy to find the exact solution to the worst-case recurrence for  $n = 2^k$ :

$$C(n) = \frac{3}{2}n - 2$$

c

The divide-and-conquer algorithm and brute-force algorithm have the same time complexity, but the divide-and-conquer algorithm make less (about 25%) elementary operations than the brute-force algorithm.

## 3 Exercises 5.1 - 5

a

$$a = 4, b = 2, d = 1. a > b^d$$

$$\text{Thus, } T(n) \in \Theta(n^{\log_2^4}) = \Theta(n^2)$$

**b**

$a = 4, b = 2, d = 2. a = b^d$   
Thus,  $T(n) \in \Theta(n^2 \log n)$

**c**

$a = 4, b = 2, d = 3. a < b^d$   
Thus,  $T(n) \in \Theta(n^3)$

## 4 Exercises 5.3 - 2

It's not correct. The correct version:

```
//Input: A binary tree T
//Output: The number of leaves in T
function LEAFCOUNTER(T)
  if then T =  $\emptyset$ 
    return 0
  else if  $T_L = \emptyset$  &  $T_R = \emptyset$  then
    return 1
  else
    return LeafCounter( $T_L$ ) + LeafCounter( $T_R$ )
  end if
end function
```

## 5 Exercises 5.3 - 5

**a**

a, b, d, e, c, f

**b**

d, b, e, a, c, f

**c**

d, e, b, f, c, a

## 6 Exercises 5.4 - 2

**First Step:**

$a1 = 21, a0 = 01, b1 = 11, b0 = 30$

$$c2 = a1 * b1 = 21 * 11, c0 = a0 * b0 = 01 * 30, c1 = (21 + 01) * (11 + 30) - (c2 + c0) = 22 * 41 - 21 * 11 - 01 * 01 * 30$$

**Second Step:**

$$\text{For } 21 * 11: c1 = (2+1)*(1+1)-(2+1) = 3, \text{ thus, } 21*11 = 2*10^2 + 3*10 + 1 = 231$$

$$\text{For } 01 * 30: c1 = (0+1)*(3+0)-(0+0) = 3, \text{ thus, } 01*30 = 0 + 3*10 + 0 = 30$$

$$\text{For } 22*41: c1 = (2+2)*(4+1)-(8+2) = 10, \text{ thus, } 22*41 = 8*10^2 + 10*10 + 2 = 902$$

**At Last:**

$$\text{The result is: } 2101*1130 = 231*10^4 + (902 - 231 - 30)*10^2 + 30 = 2,374,130$$

## 7 Exercises 5.4 - 5

$$C(n) = 2 \sum_i^n i = (n-1)^2$$

## 8 Exercises 5.4 - 6

$$C_{1,1} = M_1 + M_4 - M_5 + M_7$$

$$a_{00}b_{00} + a_{01}b_{10} = (a_{00} + a_{11})(b_{00} + b_{11}) + a_{11}(b_{10}b_{00}) - b_{11}(a_{00} + a_{01}) + (a_{01}a_{11})(b_{10} + b_{11})$$

$$C_{1,2} = M_3 + M_5$$

$$a_{00}b_{01} + a_{01}b_{11} = a_{00}(b_{01}b_{11}) + (a_{00} + a_{01})b_{11}$$

$$C_{2,1} = M_2 + M_4$$

$$a_{10}b_{00} + a_{11}b_{11} = (a_{10} + a_{11})b_{00} + a_{11}(b_{10}b_{00})$$

$$C_{2,2} = M_1 - M_2 + M_3 + M_6$$

$$a_{10}b_{01} + a_{11}b_{11} = (a_{00} + a_{11})(b_{00} + b_{11}) + a_{00}(b_{01}b_{11}) - b_{00}(a_{10} + a_{11}) + (a_{10}a_{00})(b_{00} + b_{01})$$

All above are equations.

## 9 Exercises 5.4 - 7

$$A_{00} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}, A_{01} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}, A_{10} = \begin{bmatrix} 0 & 1 \\ 5 & 0 \end{bmatrix}, A_{11} = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix},$$

$$B_{00} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}, B_{01} = \begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix}, B_{10} = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}, B_{11} = \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix}$$

$$M1 = (A_{00} + A_{11})(B_{00} + B_{11}) = \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix}$$

$$M2 = (A_{10} + A_{11})B_{00} = \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix}$$

$$M3 = A_{00}(B_{01}B_{11}) = \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix}$$

$$M4 = A_{11}(B_{10}B_{00}) = \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix}$$

$$M5 = (A_{00} + A_{01})B_{11} = \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix}$$

$$M6 = (A_{10}A_{00})(B_{00} + B_{01}) = \begin{bmatrix} 2 & 3 \\ -2 & -3 \end{bmatrix}$$

$$M7 = (A_{01}A_{11})(B_{10} + B_{11}) = \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix}$$

$$C_{00} = M_1 + M_4 - M_5 + M_7 = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

$$C_{01} = M_3 + M_5 = \begin{bmatrix} 7 & 3 \\ 1 & 9 \end{bmatrix}$$

$$C_{10} = M_2 + M_4 = \begin{bmatrix} 8 & 1 \\ 5 & 8 \end{bmatrix}$$

$$C_{11} = M_1 - M_2 + M_3 + M_6 = \begin{bmatrix} 3 & 7 \\ 7 & 7 \end{bmatrix}$$

$$C = \begin{bmatrix} 5 & 4 & 7 & 3 \\ 4 & 5 & 1 & 9 \\ 8 & 1 & 3 & 7 \\ 5 & 8 & 7 & 7 \end{bmatrix}$$