Habib University - Algorithms: Design and Analysis (CS/CE 412/471) – Spring 2025

Quiz 02 Solution

Name: ____ ID: __________ Section: ________

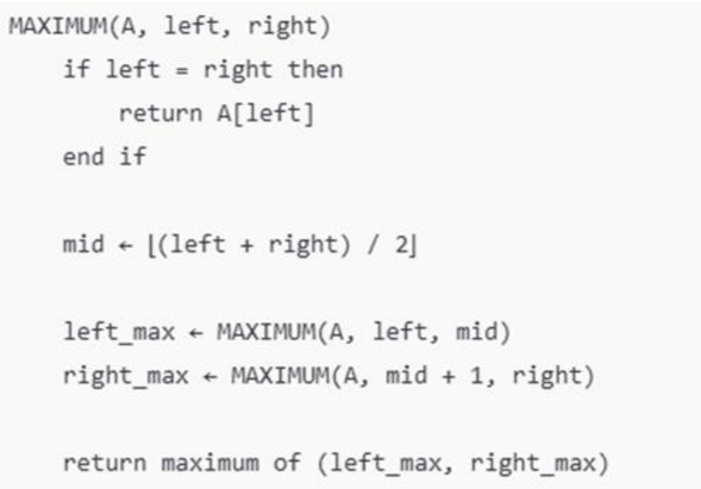
Q1. We need to design a divide-and-conquer algorithm to find the maximum element in an array.

Note: No credit if you suggest a Brute Force/non-recursive solution.

a) [1 point] State the input(s): *An array A[1..n] of n elements*

b) [1 point] State the output(s): *Maximum element in the array*

c) [5 points] Write down the algorithm *MAXIMUM* which returns the maximum of the array.



```
MAXIMUM(A, left, right)
    if left = right then
        return A[left]
    end if

    mid ← ⌊(left + right) / 2⌋

    left_max ← MAXIMUM(A, left, mid)
    right_max ← MAXIMUM(A, mid + 1, right)

    return maximum of (left_max, right_max)
```

d) [3 points] Write down the recurrence for the above algorithm:

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

e) [5 points] Find out the time complexity using Iterative/backward substitution . (back side)

T(n) = 2T(n/2) + c

T(n) = 2 [2T(n/4) + c] + c
= 4T(n/4) + c + c
= 4T(n/4) + 2c

T(n) = 4 [2T(n/8) + c] + 2c
= 8T(n/8) + c + 2c
= 8T(n/8) + 3c

…

T(n) = 2^k T(n / 2^k) + k * c

The recursion stops when n / 2^k = 1:

n / 2^k = 1 -> 2^k = n -> k = log_2 n

T(n) = 2^(log_2 n) * T(1) + c * log_2 n
= n * Θ(1) + c log_2 n
= Θ(n) + c log_2 n

The dominating term is n, so:

T(n) = Θ(n)

f) [5 points]. Now consider the following problem which needs to be solved using divide and conquer.

Given an array A[1..n], find the maximum sum of a contiguous sequence.

e.g.

1	2	3	4	5	6	7	8
-2	-5	6	-2	-3	1	5	-6

Maximum Sum = 7 (i.e. A[3]..A[7])

e.g.

1	2	3	4	5	6	7	8
1	-3	2	1	-1	3	-2	4

Maximum Sum = 7 (i.e. A[3]..A[8])

State the recurrence for this problem. Briefly state how you reached your answer.

$$\boldsymbol{T(n) = 2\,T\left(\frac{n}{2}\right) + \theta(n)}$$

Similar to finding MAX, just that the work done, f(n) is $\boldsymbol{\theta(n)}$ instead of $\boldsymbol{\theta(1)}$.