Muhammad Abbas
23K-0068

DAA Assignment 2 Date:

Q1) QuickSort Algorithm (Descending):

```
Sort_Desc (A, low, high):
    if low < high:
        p = Partition_Desc (A, low, high)
        Sort_Desc (A, low, p-1)
        Sort_Desc (A, p+1, high)

Partition_Desc (A, low, high):
    pivot = A [high]
    i = low - 1
    for j = low to j = high - 1:
        if A[j] > pivot:
            i = i + 1
            swap (A[i], A[j])
    swap (A [i+1], A [high])
    return i+1
```

Worst Case: $O(n^2)$ → Pivot always extreme element
Average Case: $O(n \log n)$
Best Case: $O(n \log n)$

Q2) 1)

```
MergeSort (A, low, high):
    if left >= right:
        return
```

→

Date:

```
mid = (low + high) / 2
MergeSort (A, low, mid)
MergeSort (A, mid + 1, high)
Merge (A, low, mid, high)


Merge(A, low, mid, high).
    n1 = mid - low + 1
    n2 = high - mid
    L [n1]
    R [n2]
    for i = 1 to n1 :
        L [i] = A [low + i - 1]
    for j = 1 to n2 :
        R [j] = A [mid + j]
    i = 1    j = 1    k = low
    while i ≤ n1 and j ≤ n2:
        if L [i] ≤ R [j]:
            A [k] = L [i]
            i = i + 1
        else:
            A [K] = R [j]
            j = j + 1
        k = k + 1
    while i ≤ n1 :
        A [k] = L [i]
        i + 1
        k + 1
    while j ≤ n2 :
        A [k] = R [j]
        j + 1
        k + 1
```

SP PAPER PRODUCT

Date:

Q4) [59, 6, 35, 12, 27, 9, 1, 18, 5, 31, 16]

59 6 35 12 27 9 | 1 18 5 31 16

59 6 35 | 12 27 9 | 1 18 5 | 31 16

59 6 35 | 12 27 9 | 1 18 5 | 31 16

59 | 6 | 35 | 12 | 27 | 9 | 1 | 18 | 5 | 31 | 16

6 59 35 | 12 27 9 | 1 18 5 | 16 31

6 35 59 | 9, 12 27 | 1 5 18 | 16 31

6 9, 12 27 35 59 | 1 5 16 18 31

1 5 6 9 12 16 18 27 31 35 59

3) Recurrence: $T(n) = 4T(n/4) + O(n)$

Q5) Using Masters Theorem: $T(n) = aT(n/b) + f(n)$

$a = 4 \quad b = 4, \quad f(n) = O(n)$

$p = \log_b a = \log_4 4 = 1$

$f(n) = O(n^p) \rightarrow O(n^1)$

$T(n) = O(n \log n)$

$\longrightarrow$

Date:

Q3)

a) Matrix Multiplication (W, R, n):

    for i=0 to n-1:
        for j=0 to n-1:
            c[i][j] = 0
            for k=0 to n-1:
                c[i][j] = C[i][j] + (W[i][k] * R[k][j])
    return C

$T(n) = O(n^3)$ → 3 loops from 1 to n

b) Strassen's Algorithm divides matrix into 2x2 sub-matrices so for a 4x4 matrix, 4 submatrices of matrix A and 4 submatrices of matrix B will be formed

let $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$

$C = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}$

$M_1 = (A_{11} + A_{22})(B_{11} + B_{22})$

$M_2 = (A_{21} + A_{22}) B_{11}$

$M_3 = A_{11}(B_{12} - B_{22})$

$M_4 = A_{22}(B_{21} - B_{11})$

$M_5 = (A_{11} + A_{12}) B_{22}$

$M_6 = (A_{12} - A_{11})(B_{11} + B_{12})$

$M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$

→

Date: ____________

$C_1 = M_1 + M_4 - M_5 + M_7$
$C_2 = M_3 + M_5$
$C_3 = M_2 + M_4$
$C_4 = M_1 - M_2 + M_3 + M_4$

$$A \times B \Rightarrow C = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}$$

c) Strassen's Algorithm performs 7 recursive multiplications of size $n/2$ and $O(n^2)$ work for arithmetic operators

$$T(n) = 7T(n/2) + O(n^2)$$

d) Using Masters Theorem:

$a = 7$ $b = 2$ $f(n) = O(n^2)$

$\log_b a = \log_2 7 = 2.807$

Since $n^{2.807}$ is a larger expression than $f(n) \to n^2$ So,

$$T(n) = O(n^{\log_2 7})$$

e) By analyzing the time complexities, $O(n^3)$ and $O(n^{\log_2 7})$ it can be concluded that strassen's algorithm is more efficient than navie approach.

Date:

Q4) $T(n) = aT\left(\frac{n}{b}\right) + f(n)$; $a \geq 1$, $b > 1$

$p = \log_b a$

Case 1 : if $\varepsilon > 0$ exists such that:

$f(n) = O(n^{p-\varepsilon})$

then recursion dominates and

$T(n) = O(n^p)$

Case II : $f(n) = O(n^p \log^k n)$; $k \geq 0$

then

$T(n) = O(n^p \log^{k+1} n)$

Case III : if $\varepsilon > 0$ exists such that:

$f(n) = O(n^{p+\varepsilon})$

then non-recursive term dominates and,

$T(n) = O(f(n))$

~~Q5) Using Masters Theorem~~ $\cancel{T(n) = aT(n/b) + f(n)}$

$\cancel{a = 4 \quad b = 4 \quad f(n) = O(n)}$

$\cancel{p = \log_b a = \log_4 4 = 1}$

Q6) I. $T(n) = 2T(n/2) + n^2$

~~Recursion~~

Date:

Recursion Tree:

n

$n/2 \quad n/2 \qquad n^2/2$

$n/4 \quad n/4 \quad n/4 \quad n/4 \qquad n^2/4$

$n/8 \quad n/8 \quad n/8 \quad n/8 \quad n/8 \quad n/8 \quad n/8 \quad n/8 \qquad n^2/8$

$\vdots$

$T(n) = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \dots + T(1)$

$T(n) = 2n^2$

$T(n) = O(n^2)$

Substitution: $T(n) = 2T(n/2) + n^2$ — (1)

$T(n/2) = 2T(n/4) + (n/2)^2$ substitution (1)

$T(n) = 2[2T(n/4) + (n/2)^2] + n^2$

$T(n) = 4T(n/4) + 2(n^2/4) + n^2$

$4T(n/4) + n^2/2 + n^2$

$T(n/4) = 2T(n/8) + (n/4)^2$ — (2)

$T(n) = 8T(n/8) + (n^2/4) + n^2/2 + n^2$

$T(n) = 2^k T(n/2^k) + n^2(1 + 1/2 + 1/4 + \dots + 1/2^{k-1})$

$\frac{n}{2^k} = 1, \; k = \log_2 n$

$T(n) = 2^{\log_2 n} \; T(1) + n^2(1 + 1/2 + 1/4 + \dots 1/2^{\log_2 n - 1})$

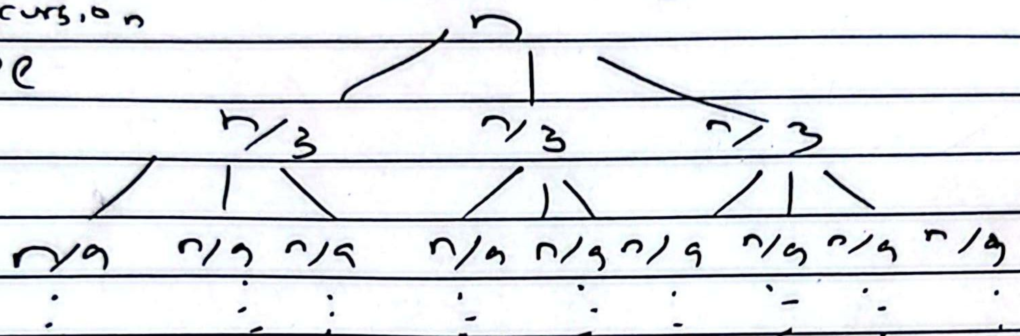
$T(n) = n + n^2(2 - 1/n) \rightarrow 2n^2 - n + nc$

$T(n) = O(n^2)$

Date:

II) $T(n) = 3T(n/3) + n\log^2 n$

Recursion Tree



$T(n) = n\log^2 n + n\log^2 n/3 + n\log^2 n/9 + \ldots$

$T(n) = n(\log^2 n + \log^2 n/3 + \log^2 n/9 + \ldots)$

$T(n) = O(n\log^2 n)$

Substitution: $T(n) = 3T(n/3) + n\log^2 n$ — ①

$T(n/3) = 3T(n/9) + n/3 \log^2(n/3)$ → substitute

$T(n) = 3(3T(n/9) + n/3 \log^2(n/3)) + n\log^2 n$

$T(n) = 9T(n/9) + n\log^2(n/3) + n\log^2 n$

$T(n) = 3^k T(n/3^k) + n(\log^2 n + \log^2(n/3) + \log^2(n/3^{k-1}))$

$n/3^k = 1, \quad k = \log_3 n$

$T(n) = 3^{\log_3 n} T(1) + n(\log^2 n + \log^2(n/3) + \ldots + \log^2(n/3^k))$

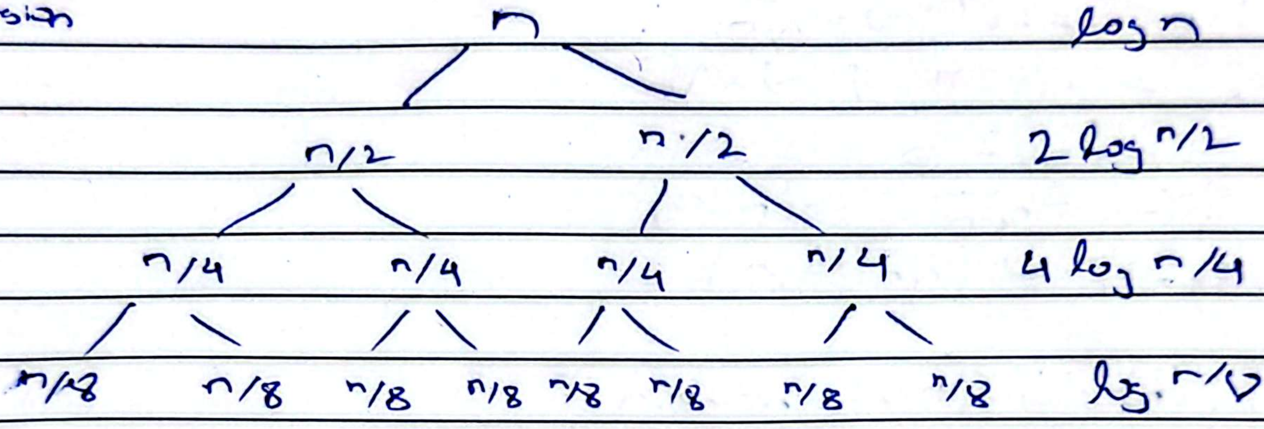
$T(n) = n + n(\log^2 n)$

$T(n) = O(n\log^2 n)$

III) $T(n) = 2T(n/2) + \log n$

Date:

Recursion



$T(n) = \log n + 2\log n/2 + 4\log n/4 + 8\log n/8 + \ldots$

$T(n) = O(n)$

Substitution: $T(n) = 2T(n/2) + \log n$ — (1)

$T(n/2) = 2T(n/4) + \log(n/2)$

$T(n) = 2(2T(n/4) + \log(n/2)) + \log n$

$T(n) = 4T(n/4) + 2\log(n/2) + \log n$

$T(n/4) = 2T(n/8) + \log(n/4)$

$T(n) = 8T(n/8) + 4\log(n/4) + 2\log(n/2) + \log n$

$T(n) = 2^k T(n/2^k) + \log n + 2\log(n/2) + 2^2\log(n/2^2) + \ldots 2^{k-1}\log(n/2^{k-1})$

$T(n) = 2n - \log n - 2$

$T(n) = O(n)$

Q7) I. $T(n) = 4T(n/2) + n^2$

$a = 4 \quad b = 2 \quad f(n) = n^2$

$p = \log_2 4 \Rightarrow 2$

$f(n) = n^2 = O(n^p)$

$T(n) = O(n^2 \log n)$ →

Date:

II $T(n) = 2T(n/4) + \sqrt{n}$

$a = 2 \quad b = 4 \quad f(n) = \sqrt{n}$

$p = \log_4 2 \Rightarrow 1/2$

$f(n) = \sqrt{n} \quad O(n^{1/2})$

$T(n) = O(\sqrt{n} \log n)$

III) $T(n) = 4T(n/2) + n^2 \log n$

$a = 4 \quad b = 2 \quad f(n) = n^2 \log n$

$p = \log_2 4 \Rightarrow 2$

$F(n) = n^2 \log n = O(n^p \log^2 n)$

$T(n) = O(n^2 \log^2 n)$

Q8) I) $T(n) = 4T(n/2) + n^2$

Guess 1 : $T(n) = O(n)$

$T(n) \leq c.g(n)$

$T(n) = 4T(n/2) + n^2 \leq 4 . c(n/2) + n^2$

$4T(n/2) + n^2 \leq n^2 + 2cn$

For $T(n) \leq cn$:

$2cn + n^2 \leq cn$

$n^2 + nc \leq 0$

Guess 1 is false

Guess 2 : $T(n) = O(n^2)$

$T(n) \leq c.g(n)$

$+ \frac{n^2}{2}$

$T(n) = 4T(n/2) + n^2 \leq 4c(n/2$

$4T(n/2) + n^2 \leq cn^2 + n^2$

$n^2 + \cancel{cn} \leq \cancel{cn}$

$n^2 \leq 0$

Guess 2 is also false