DAA

Date ______ M T W T F S S

ASSIGNMENT 2

Q1:

```
QuickSort (arr, low, high)
{
    if low < high
    {
        p = Partition (arr, low, high)
        QuickSort (arr, low, p-1)
        QuickSort (arr, p+1, high)
    }
}
Partition (arr, low, high)
{
    pivot = arr [high]
    i = low - 1
    for j = low, j < high
    {
        i = i + 1
        swap arr[i], arr[j]
    }
    swap arr[i+1], arr[high]
    return i+1
}
```

⇒ Best Case: $O(n \log n)$

⇒ Average Case: $O(n \log n)$

⇒ Worst Case: $O(n^2)$

Q2.

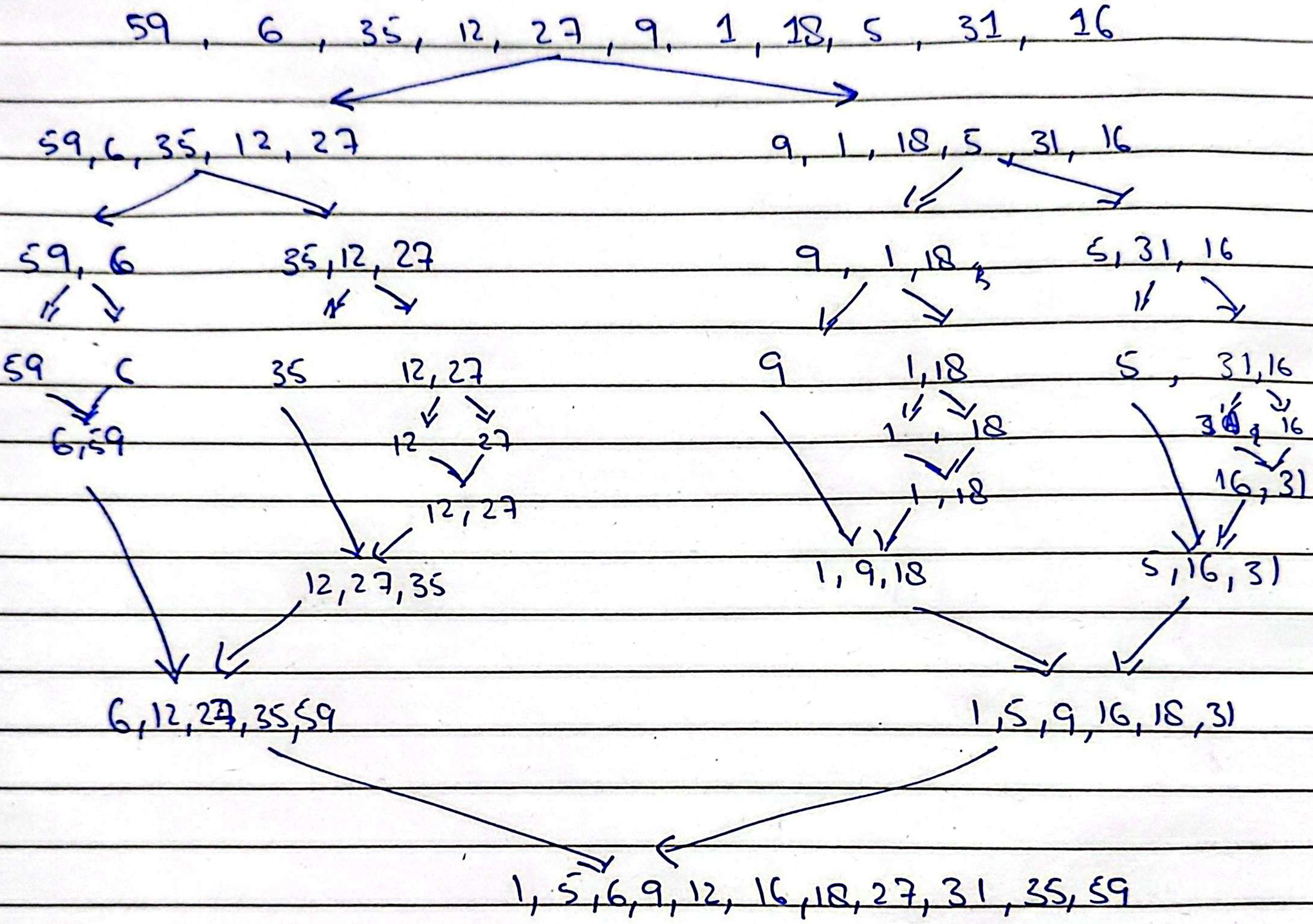
```
MergeSort (arr, left, right)
{
    if left < right
    {
        mid = (left +right)/2
        MergeSort (arr, left, mid)
        MergeSort (arr, mid+1, right)
        Merge( arr, left, mid, right)
    }
}
Merge(arr, l, m, r)
{
    n1 = m - l + 1
    n2 = r - m
    L[n1] , R[n2]
    for i = 1 to i = n1
    {
        L[i] = arr[l + i - 1]
    }
    for j = 1 to j = n2
    {
        R[j] = A[mid + j]
    }
```

Page No. ______ Signature ______

Date

M	T	W	T	F	S

```
i = 1 , j = 1 , k = l
while i ≤ n1 and j ≤ n2
{
    if L[i] ≤ R[j]
    {
        arr[k] = L[i]
        i = i + 1
    }
    else
    {
        arr[k] = R[j]
        j = j + 1
    }
    k = k + 1
}
while i ≤ n1 :
{
    arr[k] = L[i]
    i = i + 1
    k = k + 1
}
while j ≤ n2
{
    arr[k] = R[j]
    j = j + 1
    k = k + 1
}
}
```



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

⇒ $a = 4$, $b = 4$, $f(n) = O(n)$

↳ $\log_4 4 = 1$

↳ $f(n) = \Theta(n^p) \rightarrow O(n^2)$

↳ $T(n) = O(n \log n)$

Date
M	T	W	T	F	S	S

Q3: (a)

```
Mul-Matrix (A, B, n)
{
    for i=0, to i<n                ---> n
    {
        for j=0 to j<n             ---> n (n-1)
        {
            C[i][j] = 0
            for k=0 to k<n         ---> n (n-1)(n-1)
            {
                C[i][i] += A[i][k] * B[k][i]
            }
        }
    }
}
```

$\Rightarrow T(n) = O(n^3)$ {There are three loops}.

Signature

Date ______ M T W T F S S

(b)

$M_1 = (A_{11} + A_{22}) * (B_{11} + B_{22})$
$M_2 = (A_{21} + A_{22}) * B_{11}$
$M_3 = A_{11} * (B_{12} - B_{22})$
$M_4 = A_{22} * (B_{21} - B_{11})$
$M_5 = (A_{11} + A_{12}) * B_{22}$
$M_6 = (A_{21} - A_{11}) * (B_{11} + B_{12})$
$M_7 = (A_{12} - A_{22}) * (B_{21} + B_{22})$

$\Rightarrow C_{11} = M_1 + M_4 - M_5 + M_7$
$\Rightarrow C_{12} = M_3 + M_5$
$\Rightarrow C_{21} = M_2 + M_4$
$\Rightarrow C_{22} = M_1 - M_2 + M_3 + M_6.$

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

$\Rightarrow$ 7 recursive multiplications of size $n/2$
$\Rightarrow$ $O(n^2)$ for additions/subtractions.

(d) $a = 7, b = 2, f(n) = O(n^2)$
$\Rightarrow \log_2 7 = 2.807$
$\Rightarrow$ since, $n^{2.807}$ is larger than $f(n) = n^2$:
$T(n) = O(n^{\log_2 7})$

(e) By analyzing the time complexities, it can be concluded that strassen's algorithm is asymptotically faster

Date ____
M T W T F S S

Q4.

$\Rightarrow$ $T(n) = aT\left(\frac{n}{b}\right) + f(n)$

$\Rightarrow$ $a \geq 1$, $b > 1$

$\Rightarrow$ $p = \log_b a$

$\Rightarrow$ Case 1:

$\varepsilon > 0$, such that

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

then recursion dominates

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

$\Rightarrow$ Case 2:

$f(n) = O(n^p \log^k n)$, $k \geq 0$

then,

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

$\rightarrow$ Case 3:

$\varepsilon > 0$, such that

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

then non-recursive term dominates,

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

Date
M	T	W	T	F	S

Q5:

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

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

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

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

Q7.

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

$\Rightarrow p = \log_2 4 = 2$

$\Rightarrow f(n) = n^2 \Rightarrow$

$\Rightarrow O(n^p)$

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

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

$\Rightarrow p = 1/2$

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

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

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

$\Rightarrow p = 2$

$\Rightarrow f(n) = n^2 \log n \Rightarrow O(n^p \log n)$

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

Q8:

$\Rightarrow$ Guess 1: ~~For~~ $T(n) = O(n)$

$\hookrightarrow$ $T(n) \leq c \cdot g(n)$

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

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

$\hookrightarrow$ For $T(n) \leq cn$:

$2cn + n^2 \leq cn$

$n^2 + nc \leq 0$

$\hookrightarrow$ Guess 1 is incorrect.

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

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

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

$n^2 + cn^2 \leq cn^2$

$n^2 \leq 0$

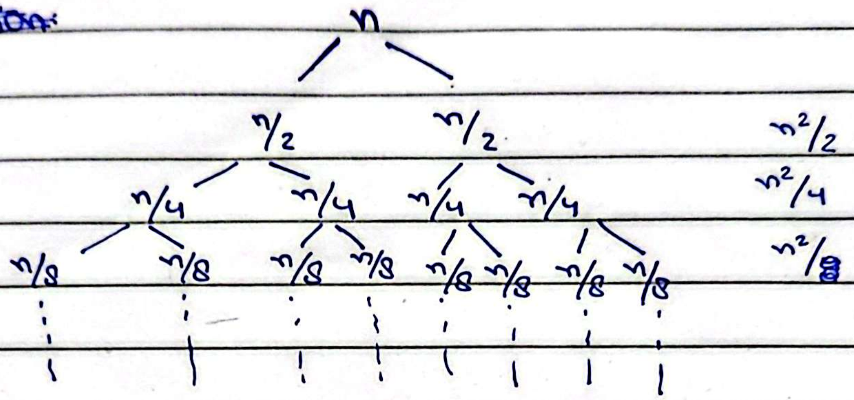
$\hookrightarrow$ Guess 2 is also incorrect.

Date
M	T	W	T	F	S	S

Q6:

1)

~~→ Substitution~~



$\Rightarrow \quad T(n) = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \frac{n^2}{8} + \dots + T(1)$

$= n^2\left(1 + 1/2 + 1/4 + 1/8 + \dots\right)$

$= 2n^2$

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

⇒ Iterative approach:

↳ $T(n) = 2T(n/2) + n^2$ — ①

↳ $T(n/2) = 2T(n/4) + (n/2)^2$ — ②

↳ Substituting ② into ①:

↳ $T(n) = 2\left[2T(n/4) + (n/2)^2\right] + n^2$

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

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

↳ $T(n/4) = 2T(n/8) + (n/4)^2$ — ③

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

↳ $T(n) = 2^k T(n/2^k) + n^2\left(1 + 1/2 + 1/4 + 1/8 + \dots + 1/\log_2^{n-1}\right)$

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

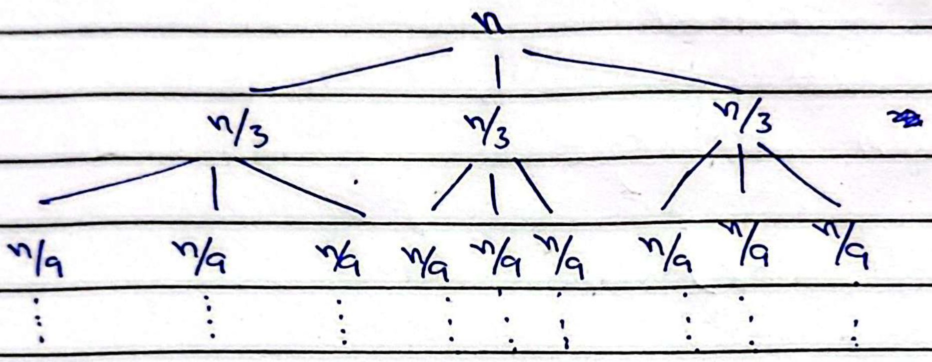
$\hookrightarrow T(n) = 2^{\log_2 n} T(1) + n^2\left(1 + 1/2 + 1/4 + \ldots + 1/2^{\log_2 n - 1}\right)$

$T(n) = 2n^2 \mp n + nc$

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

2)

$\Rightarrow$ Recursion:



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

$= n\left(\log^2 n + \log^2 n/3 + \log^2 n/9 + \cdots\right)$

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

⇒ Substitution:

↳ $T(n) = 3T(n/3) + n\log^2 n$ — ①

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

↳ ② → ①

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

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

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

↳ $\frac{n}{3^k} = 1, \quad k = \log_3 n$

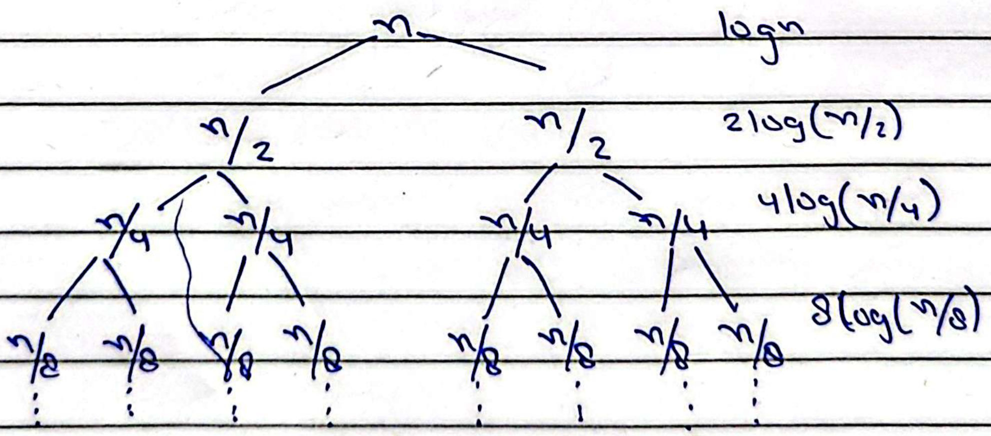
↳ $T(n) = 3^{\log_3 n}\, T(1) + n\left(\log^2 n + \log^2(n/3 + \dots + \log^2(n/3^{k-1})\right)$

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

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

3)

⇒ Recursion:



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

$= O(n)$

Date ______

M	T	W	T	F	S	S

⇒ Substitution:

↳ $T(n) = 2T(n/2) + \log n$

↳ $T(n/2) = 2T(n/4) + \log(n/2)$

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

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

↳ $T(n/4) = 2T(n/8) + \log(n/4)$

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

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

↳ $T(n) = 2^k T(n/2^k) + \log(n) + 2\log(n/2) + 2^2\log(n/2^2)$

$+ \dots + 2^{k-1}\log(n/2^{k-1})$

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

↳ $T(n) = O(n)$

Signature ______