Page No.
Date / 20

Q-1

```
int sum = 0;
for(int i=1; i<=n; i++){
    for(int j=1; j<=1; j++){
        sum++;
    }
}
```

— output = 15

Soln $O(n^2)$

Improve

```
int sum = 0;
for(int i=1; i<n; i++) {
    sum = sum + i;
}
~~Sout~~ System.out.println(sum);
```

— output = 15

Time complexity = $O(n)$

Page No.
Date / 20

Q.2 Find the value of $T(2)$ for the recurrence relation $T(n) = 3T(n-1) + 12n$ given that $T(0) = 5$

Soln $T(2) = ?$, $T(0) = 5$

$$T(n) = 3T(n-1) + 12n \quad \text{---(1)}$$

$$T(n-1) = 3T(n-2) + 12(n-1) \quad \text{---(2)}$$

put in eq---(1)

$$T(n) = 3[3T(n-2) + 12(n-1)] + 12n$$

$$T(n) = 3^2 T(n-2) + 3 \times 12(n-1) + 12n$$

$$T(2) = 9T(0) + 3 \times 12(1) + 12 \times 2$$

$$= 9 \times 5 + 36 + 24$$

$$= 45 + 36 + 24$$

$$T(2) = 105$$

Page No.
Date / /

Q-3 Give a recurrence relation, solve it using a substitution method.

$T(n) = T(n-1) + c$

Soln $T(n) = T(n-1) + c$ — (1)

$T(n-1) = T(n-2) + c$

put in eq-(1)

$T(n) = T(n-2) + 2c$ — (2)

$T(n-2) = T(n-3) + c$

put in eq-(2)

$T(n) = T(n-3) + 3c$ — (3)

$T(n) = T(n-k) + kc$

$T(1) = 1$

$n - k = 1$ $\quad \therefore T(n-k) = 1$

$k = n$

$T(n) = 1 + nc$

$O(n)$

Q-4 Give a recurrence relation:

$T(n) = 16T(n/4) + n^2 \log n$

find time complexity using master theorm.

Soln Standard eqn:

$$T(n) = aT\left(\frac{n}{b}\right) + \theta(n^K \log^P n)$$

Compare with eqn.

$a = 16, \quad b = 4, \quad K = 2, \quad P = 1$

Case-1 $a > b^K$

$16 > 4^2$ ✓ — $16 \geq 16$ ✗

Condition satisfy

Page No.
Date / /

Ca-2

$a = b^k$ —— Case-2

$16 = 16$

$p = \frac{1}{2}$ — $p > -1$

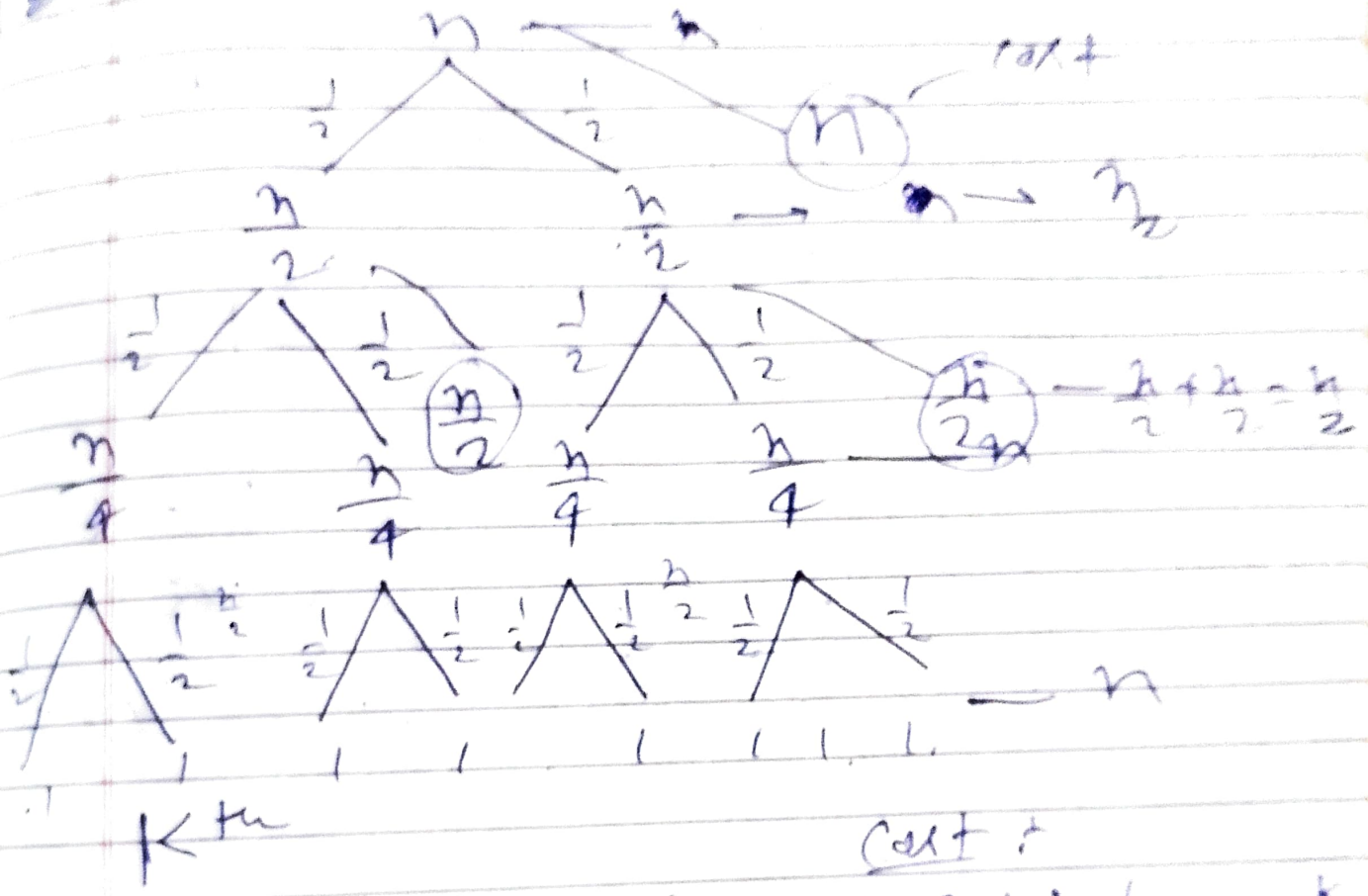
$T(n) = O(n^{\log_4 16} \log^{1+1} n)$

$= O(n^{\log_4 16} \log^2 n)$

Q. Solve recurrence relation using

Give the following recurrence relation using tree method.

$T(n) = 2T(n/2) + n$, cost



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

K^{th}

$\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log_2 n$

Cost :

$n + n + n + \cdots k$

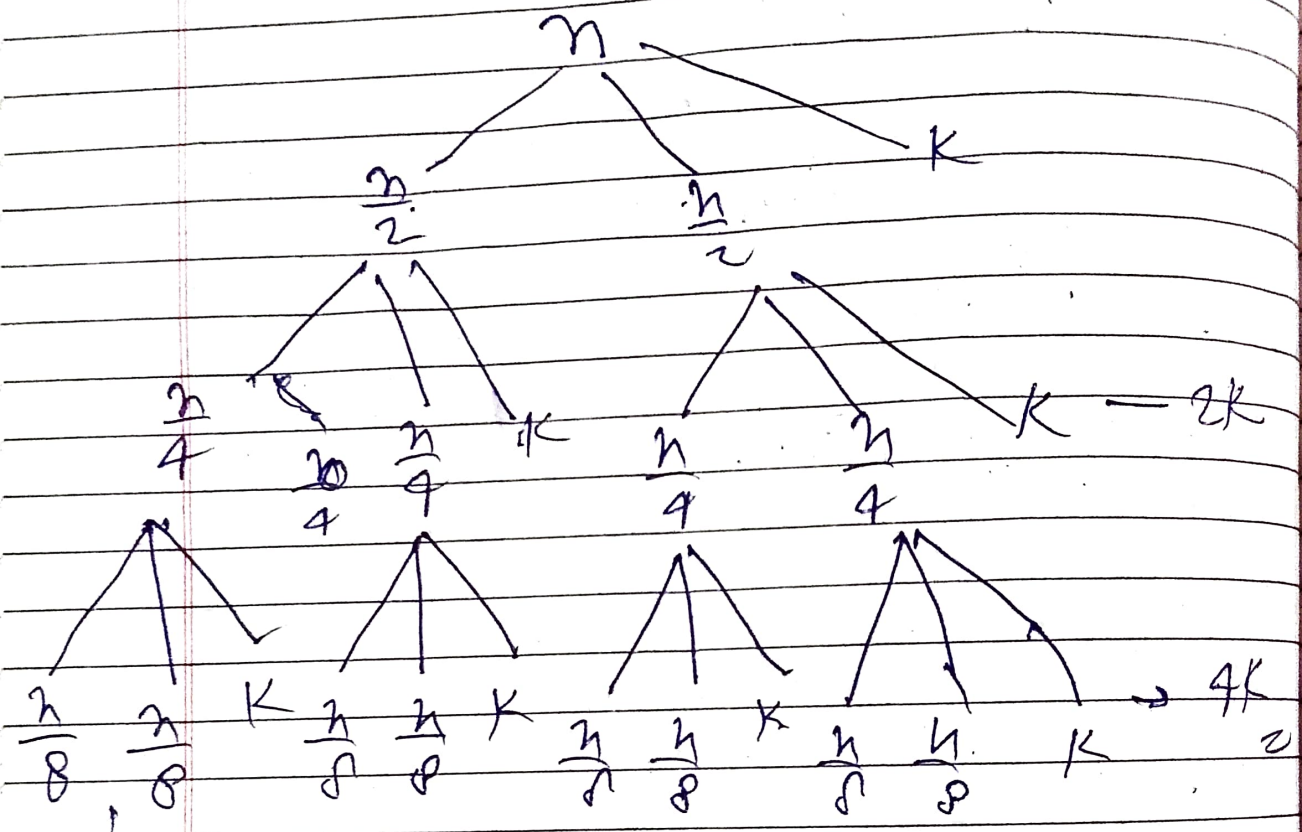
kn

$\log_2 n \times n$

$O(k \cdot n \log_2 n)$

Page No.
Date / 20

(Q-6) $T(n) = 2T(n/2) + K$ using recurrence tree method.



$$\frac{n}{2^c} = 1$$

$$n = 2^c$$

$$c = \log_2 n$$

$$K + 2K + 4K \text{ ---}$$

$$K(2^0 + 2^1 + 2^2 \text{ --- } 2^c)$$

$$K(2^0 + 2^1 + 2^2 \text{ --- } 2^{\log_2 n})$$

GP series

$$S_{\infty} = \frac{2^0 \left(2^{\log_2 n} - 1\right)}{2 - 1}$$

$$= \frac{1\,(n-1)}{}$$

constant

$$= K(n-1)$$

$$= O(n)$$