Name- Binay Singh
Section- A
Subject - DAA
Roll no :- 1961033
Course - B. Tech (CSE)

Tutorial sheet 1

Ans-1) (3) $O(N+M)$ time
$O(1)$ space

Ans-2 $T(n) = O(n)$, space $O(1)$

Ans-3 $T(n) = O(\log_2 n)$, space $O(1)$

Ans-4
```
int sum = 0, i;
for (i=0, i*i<n, i++)
{
    sum += i;
}
```
$= n + (n-1) + (n-4) + (n-9) + \cdots (n-k)$
$= n(n+k) - (1^2 + 2^2 + 3^2 + \cdots k^2)$
$= \sqrt{n}$
$i^2 < n$
$i < \sqrt{n}$
$T(n) = O(\sqrt{n})$, space $O(1)$

Ans-5
```
int j=1, i=0
while (i<=n)
{
    i = i+j;
    j++;
```

P. No.
Dt.

(2)

}

0 <= n 1

1 <= n 1

3 <= n

(0, 1, 3, 6, 10, 15, 21 --- n)

k term

$$k^{th} \text{ term} = \frac{(k * (k+1))}{2}$$

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

$$k^2 + k - 2n = 0$$

$$k = \frac{-1 \pm \sqrt{k^2 + 8n}}{2}$$

$$= \frac{\sqrt{8n+1} + 1}{2}$$

$$k = \frac{\sqrt{8n+1}}{2}$$

$$= \frac{\sqrt{8n}}{2} = \sqrt{n}$$

$T(n) = \sqrt{n}$ space - O(1)

Ans-6

```
void Recursion (int n) → T(n)
{
    if (n == 1) return;
    recursion (n-1)  → T(n-1)
    print(n);
    recursion (n-1); → T(n-1)
}
```

$$T(n) = \begin{cases} 1 & n=1 \\ 2T(n-1)+1 & n>1 \end{cases}$$

$T(n) = 2T(n-1)+1$ — 1

$T(n-1) = 2T(n-2)+1$
$T(n) = 2(2T(n-2)+1)+1$
$T(n) = 4T(n-2)+(1+2)$ — 2

$T(n-2) = 2(T(n-3)+1$
$T(n) = 4(2T(n-3)+1)+(1+2)$
$T(n) = 8T(n-3)+(1+2+4)$ — 3

$T(n) = 8[2T(n-4)+1]+(1+2+4)$
$T(n) = 16T(n-4)+(1+2+4+8)$ — 4

$T(n) = 2^k T(n-k) + \underbrace{(1+2+4+8+\cdots)}_{k \text{ times}}$

$T(n-k) = T(1)$
$k = n-1$

$T(n) = 2^{n-1} T(1) + \underbrace{(1+2+4+8 \cdots)}_{(n-1) \text{ times}}$

$T(n) = \frac{2^n}{2} + \underbrace{(1+2+4+8 \cdots)}_{(n-1) \text{ times}}$

$S_n = \frac{a(r^n-1)}{r-1}$ $a=1, r=2, n=n-1$

$T_n = \frac{2^n}{2} + \left(\frac{2^{n-1}-1}{1}\right)$

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

$$T(n) = \cancel{2}\left(\frac{2^n}{\cancel{2}}\right) - 1$$

$$T(n) = 2^n - 1$$

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

Ans-7 It is a Binary search Algorithm.

$$T(n) = \log_2 n$$

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

by using Masters method (can't be solved)

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

so $a = 1$

$b = 2$

$f(n) = 1$

$$c = \log a = \log_2 1 = 0$$

$$0 \cong 1$$

$$n^0 = f(n) = 1$$

$$n^c = f(n)$$

$$T(n) = O(\log_2 n)$$

Ans-8

$T(1) = 1$

1). $T(n) = T(n-1) + 1$ — (1)

$T(n) = T(n-2) + 2$ — (2)

$T(n) = T(n-3) + 3$ — (3)

$T(n) = T(n-k) + k$ — (4)

$n - k = 1$

$k = n - 1$

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

$T(n) = n$

$T(n) = O(n)$

2) $T(n) = T(n-1) + n$ — (1)

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

$T(n) = T(n-2) + (n + (n-1))$ — (2)

$T(n) = T(n-3) + (n + (n-1) + (n-2))$ — (3)

$T(n) = T(n-k) + (n + (n-1) + (n-2) \cdots (n-k))$

$T(n-k) = T(1)$

$n = k + 1$

$k = n - 1$

$T(n) = T(1) + (n + (n-1) + (n-2) \cdots (n-(n-1)))$

$T(n) = 1 + (n + (n-1) + (n-2) + \cdots 1)$

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

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

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

Ans-8

3 $T(n) = T(n/2) + 1$ — 1

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

$T(n) = T(n/4) + 2$ — 2

$T(n/4) = T(n/8) + 1$

$T(n) = T(n/8) + 3$ — 3

$T(n) = T\left(\frac{n}{2^k}\right) + k$ — 4

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

$$2^k = n$$

$$k = \log_2 n$$

$$T(n) = T(1) + \log_2 n$$

$$T(n) = O(\log_2 n)$$

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

$c = 1$

$n^c = n$

$f(n) = 1$

$n^c > f(n)$

$T(n) = \theta(n)$

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

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

6) $T(n) = 3T(n-1), \quad T(0) = 1$

$T(n) = 3(T(n-1)) \quad - (1)$

$T(n-1) = 3(T(n-2))$

$T(n) = 9T(n-2)$

$T(n) = 3^3(T(n-3))$

$T(n) = 3^k(T(n-k))$

for $n-k = 0$

$n = k$

$T(n) = 3^n(0)$

$T(n) = 3^n$

$T(n) = O(3^n)$

$$T(n) = \begin{cases} 1, & n <= 2 \\ T(n), & n > 2 \end{cases}$$

7) $T(n) = T(\sqrt{n}) + 1 \quad - (1)$

$T(\sqrt{n}) = T(n^{1/4}) + 1 \quad (2)$

$T(n) = T(n^{1/8}) + 3 \quad - (3)$

$T(n) = T(n^{1/8}) + 3$ — ③

$T(n) = T(n^{1/2}) + k$

for $T((\sqrt{n})^{1/k}) = T(2)$

$n^{1/2k} = 2$

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

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

$2^k = \log n$

$2^k = \log n$

$k = \log_2(\log n))$

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

⑧ $T(n) = T(\sqrt{n}) + n$

$T(\sqrt{n}) = T(n^{1/4}) + \sqrt{n}$

$T(n = T(n^{1/4}) + (n + \sqrt{n})$

$T(n) = T(n^{1/4}) + (n + \sqrt{n})$

$T(n) = T(n^{1/8}) + (n + \sqrt{n} + n^{1/4})$

$T(n) = T(n^{1/8}) + (n + \sqrt{n} + n^{1/4})$

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

k terms)

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

$$\frac{1}{2^k} = 1/\log(n)$$

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

$$k = \log(\log(n))$$

$$T(n) = 1 + \left(n + \sqrt{n} + \sqrt{n}\sqrt{n} + \cdots \right)$$

k terms.

$T(n) = 1 + \Big($ G.P. $a = n$, $r = \sqrt{n}$, no. of terms $\cdots \Big)$

$T(n) = 1 + \Big($ G.P. $a = n$, $r = \sqrt{n}$, no. of terms $= k \Big)$

$$T(n) = 1 + \left(\frac{n(\sqrt{n})^k - 1}{k - 1} \right)$$

$$T(n) = 1 + n \left(\frac{(\sqrt{n})^{\log\log(n)} - 1}{\log\log(n) - 1} \right)$$

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

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

Ans-9

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

0, 1, 2, ----- n

so. $T(n) = O(n)$, space $O(1)$

Ans-10

$O(N * (N, N-1, --- 1))$

$O\left(N * \left(\frac{N+1}{2}\right)\right)$

(4.) $O(N * N)$

Ans-11

$O\left(\frac{n}{2} * (\log_2 N)\right)$

$O(n \log n)$

Ans-12 (2) X will always be a better choice for large input.

Ans-13 (4) $O(\log N)$

(14) $T(n) = 7\left(T\left(n/2\right)\right) + (3n^2 + 2)$

$f(n) = 7\left(T\left(n/2\right)\right) + (3n^2 + 2)$

$f(n) = 3n^2 + 2$

$a = 7$

$b = 2$

$c = \log_b a = \log_2 7 = 2.807$

$n^c = n^{2.8} \approx n^{2.8}$

$f(n) = 3n^2 + 2$

so $n^c > f(n)$

so $T(n) = \Theta(n^{2.8})$ or c $O(n^{2.8})$
= $\quad$ $O(n^{2.8})$
$\quad$ d $O(n^3)$

Ans-15

$f_1(n) = n^{\sqrt{n}}$

$f_2(n) = 2^n$

$f_3(n) = (1.000001)^n$

$f_4(n) = n(10 * 2n/2)$

a) $f_2(n) > f_4(n) > f_3(n) > f_1(n)$

Ans-16

$f(n) = 2^{2n}$

$\log f(n) = 2n \log_2 2$

$\log f(n) = 2n$

or

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

$n(2^n) =$

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

$c = 1$

$n^c = n$

$n^2 > n$

$f(n) > n^c$

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

Ans-18 $O(\log n)$ (It a G.C.D function)

Ans-19 $T(n) = O(n^2 + n)$

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