Date. __/__/____

Name- Anuk Prasad

Sec - A

Roll no - 1961021

Subject - DAA

Course - B.Tech (C.S.E)

Tutorial Sheet - 1

CC.S.C

Tutorials Sheet 1

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

Ans 2. $T(n) = O(n^2)$, 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++;
  }
```

$0 <= n$ 1
$1 <= n$ 1
$3 <= n$

$$(0, 1, \underbrace{3, 6, 10, 15, 21, \ldots n}_{K \text{ term}})$$

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

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

$$K^2 + K = 2n$$

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

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

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

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

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

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

Ans 6 - void Recursion(int n) $\longrightarrow T(n)$

{

if (n == 1) return;

recursion (n - 1) $\longrightarrow T(n-1)$

print.(n); $\longrightarrow 1$

recursion (n-1); $\longrightarrow T(n-1)$

}

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

$$T(n) = 2T(n-1) + 1 \quad — ①$$

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

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

$$T(n) = 4T(n-2) + (1+2) \quad — ②$$

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

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

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

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

$$T(n) = 16T(n-4) + (1+2+4+8) \quad — ④$$

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

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

$$k = n-1$$

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

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

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

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

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

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

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

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

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

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

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 meathod (can't be solved)

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

so $a = 1$
$b = 2$
$f(n) = 1$

T(n) = $c = \log_b a = \log_2 1 = 0$

$\simeq 1$

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

$\therefore \quad n^c = f(n)$

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

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

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

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

$n = k+1$

$k = n-1$

$T(n) = T(1) + (n + (n-1) + (n-2) - \cdots (\underbrace{\cancel{n}-\cancel{n}+1-1}_{0}))$

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

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

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

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

Ans 8
(Ans 3)-

$$T(n) = T(n/2) + 1 \quad — (1)$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + 1$$

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

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{8}\right) + 1$$

$$T(n) = T\left(\frac{n}{8}\right) + 3 \quad — (3)$$

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

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

$$2^k = n$$

$$k = \log_2 n$$

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

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

Ans 8
(Ans 4) $T(n) = 2T\left(\frac{n}{2}\right) + 1.$

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

$f(n) = n \log n$

$n^c > f(n)$
$T(n) = \Theta(n)$

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

Ans 8
(Ans 5) $T(n) = 2T(n-1) + 1$

$T(n) = 64\,T\left(\frac{n}{2}\right) + n^6$

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

$f(n) = n^6$

Ans 8
(Ans 6) $T(n) = 3T(n-1)$, $T(0) = 1$

$c = \log_b a$
$c = \log_2 64$
$c = \log_2 2^6$
$c = 6$
$n^6 = n^6$
$n^6 \log n$

$T(n) = 3(T(n-1)$ — ①
$T(n-1) = 3T(n-2)$
$T(n) = 9T(n-2)$
$T(n) = 3^3 T(n-3)$

$T(n) = 3^k T(n-k)$
for $n-k=0$
$n=k$

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

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

$T(n) = 3^n$

$c = 1$
$\log(n)$
$n > \log$

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

$T(n) = 7\,T(n) \;;\; n > 2$

Ans 8

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

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

$T(n) = T(n^{1/4}) + 2$ —(2)

$T(n) = T(n^{1/8}) + 3$ —(3)

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

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

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

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

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

$2^k = \log n$

$2^k = \log_n$

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

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

Scanned By Scanner Go

Ans 8 -

(8)- $T(n) = T(\sqrt{n}) + n$

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

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

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

$T(n) = T\left(n^{\frac{1}{2^k}}\right) + (\underbrace{n + n^{1/2} + n^{1/4} + \dots}_{K \text{ terms}})$

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

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

$2^K = \log(n)$

$K = \log(\log(n))$

$T(n) = 1 + (\underbrace{n + \sqrt{n} + \sqrt{n}\sqrt{n} + \dots}_{K \text{ terms}})$

$T(n) = 1 + \left(\begin{array}{c} G.P \quad a = n \\ r = \sqrt{n} \\ \text{No. of terms} = K \end{array} \right)$

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

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

$T(n) = n \cdot \log\log(n)$ {by neglecting other values

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

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 -

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

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

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

Ans 11 - $O\left(\frac{n}{2} * (\log_2 N)\right)$

$O(n \log n)$ //

PAGE NO
DATE / /

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

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

Ans 14 -

$$T(n) = 7\left(T\left(\frac{n}{2}\right)\right) + (3n^2 + 2)$$

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

$$a = 7$$

$$b = 2$$

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

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

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

so $n^c > f(n)$

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

Ans 15 - $f_1(n) = n^{\sqrt{n}}$

$f_2(n) = 2^n$

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

$f_4(n) = n(10 * 2^{n/2})$

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

$\log f_1 = \sqrt{n} \log n$ ($\sqrt{500} \log 500 = 11.18$)

$\log f_3 = n \log(1.0000001) = \sqrt{n}\sqrt{n} \log(1.000001)$

($500 \log(1.000001) = 0.0$

Scanned By Scanner Go

Ans 16 - $f(n) = 2^{2n}$

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

$\log f(n) = 2n$

or

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

$\Omega(2^n)$

Ans 17 - $T(n) = 2T\left(\frac{n}{2}\right) + n^2$

$c = 1$

$n^c = n$

$n^2 > n$

$f(n) > n^c$

$T(n) = \Theta(n^2)$

Ans 18 - $O(\log N)$ [Its a G.C.D function where n keeps on decreasing by $n/2$]

Ans 19 -

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

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