Tutorial-2 (DAA)

Name → Manan Sharma
Sec → H
Class Roll No → 34
University Roll No → 2016841

Soln 1 →

```
void fun(int n)
{ int j=1, i=0;
  while (i<n)
  { i=i+j;
    j++
} }
```

$j=1, \; i=0+1$
$j=2, \; i=0+1+2$
$j=3, \; i=0+1+2+3$
⋮
Loop ends when $i > n$
$0+1+2+3 -- n > n$
$\frac{K(K+1)}{2} > n$
$K^2 > n$
$K > \sqrt{n}$ $\quad O(\sqrt{n})$

Soln 2 → Recurrence relation for Fibonacci Series:

$T(n) = T(n-1) + T(n-2) \cdot \quad T(0) = T(1) = 1$

• if $T(n-1) \approx T(n-2)$

(L.B) $T(n) = 2T(n-2)$
$= 2\{2T(n-4)\} = 4T(n-4)$
$= 4(2T(n-6))$
$= 8(T(n-6))$
$= 8(2T(n-8))$
$= 16T(n-8)$

$T(n) = 2^K T(n-2K)$ ← ....

$n-2K = 0$
$n = 2K \Rightarrow K = n/2$

$T(n) = 2^{n/2} T(0)$
$= 2^{n/2}$
$T(n) = \Omega(2^{n/2})$

• if $T(n-2) \approx T(n-1)$

$T(n) = 2T(n-1)$
$= 2(2T(n-2)) = 4T(n-2)$
$= 4(2T(n-3)) = 8T(n-3) = 2^K T(n-K)$

$n-K=0$
$\boxed{K=n}$

$T(n) = 2^K \times T(0) = 2^n = T(n) = O(2^n)$ (U.B)

Soln 3 → • $O(n \log n)$ ⇒

```
for(int i=0; i<n; i++){
  for(int j=1; j<n; j=j*2)
  { // some O(1)
  }
}.
```

• $O(n^3)$ →

```
for(int i=0; i<n; i++)
{
  for(int j=0; j<n; j++)
  {
    for(int k=0; k<n; k++)
    { //some O(1)
    }
  } }
```

• $O(\log(\log n))$ →

```
for(int i=0; i<=n; i=i*2)
{ for(int j=0; j<=n; j=j*2)
  {

    { //some O(1)
  } }
```

Soln 4 → $T(n) = T(n/4) + T(n/2) + cn^2$

Lets assume $T(n/2) >= T(n/4)$

So, $T(n) = 2T(n/2) + cn^2$

Applying Master's Theorem ($T(n) = aT(\frac{n}{b}) + f(n)$.)

$a=2, b=2 \qquad f(n)=n^2$

$c = \log_b a = \log_2 2 = 1$

$n^c = n$

Compare n^c and $f(n) = n^2$

$f(n) > n^c$ So, $T(n) = \Theta(n^2)$

Soln 5 →

```
int fun(int n)
{ for(int i=1; i<=n; i++)
  { for(int j=1; j<n; j+=i)
    { // some O(1)
    }
  }
}
```

$j=1$
$j=2$
$i=1$ — $j=3$ — n times
$j=n$

$i=2$ — $j=1$ — Loop ends when $j>n$
$j=3$ $1+3+5+7 \ldots k > n$
$j=5$ $k > n/2$.
n times.
$i=3$ — $k > n/3$
$i=4$ — $k > n/4$ $i=n$

So Time complexity $= O(n^2+n^2+n^2 \ldots) = O(n^2)$

Soln 6] for (int $i=2$; $i<=n$; $i=$ Pow(i, k))
{ // some O(1)
}
Complexity of Pow(i, k) — $O(\log n) = \log(k)$

$i=2$
$i=2^k$
$i=2^{k^2}$
$i=2^{k^n}$

Loop ends when $i>n$

$2^{k^n} > n$

$\log(2^{k^n}) > \log n$

$k^n \log 2 > \log n$

$k^n > \log n$

$\log(k^n) > \log(\log n)$

$n \log k > \log(\log n)$

$n > \frac{\log(\log n)}{\log(k)}$

$T_c = O(\log(\log n))$.

soln7 → In quick sort 99 to 1 where pivot is where front or end always

so, $T(n) = T\left(\frac{99n}{100}\right) + T(n/100) + O(n)$

$T(n) = T\left(\frac{99n}{100}\right) + T(n/100) + O(n)$

$T(n)$

$T\left(\frac{99n}{100}\right)$ $\quad T\left(\frac{n}{100}\right)$ —— $T\left(\frac{kn}{100}\right)$

$T\left(\frac{(99)^2}{(100)^2} \times n\right)$ $\quad T\left(\frac{99n}{100}\right)^2$ $\quad T\left(\frac{99n}{100}\right)$

$n = (99/100)^k$

$\log n = k \log 99/100$

$k = \log n \frac{100}{9}$

$TC = n * \log_{\frac{100}{99}}(n)$

soln8 → a) $1\omega < \log n < \sqrt{n} < n < \log(\log n) < n \log n$
$< \log n! < n! < n^2 < \log^{2n} < 2^n < 2^{2n} < 4^n$

b) $1 < \sqrt{\log n} < \log n < 2 \log n < \log 2N < 2N < 4N <$
$\log(\log N) < N \log N < \log N! < N! < N^2 < 2 \times 2^N$

c) $96 < \log_8 N < \log_2 N < n \log_6 N < n \log_2 N < \log N!$
$< N! < 5N < 8N^2 < 7N^3 < 8^{2n}$