Tutorial-2 DAA

Ans-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 $i>=n$

$0+1+2+3 \ldots n > n$

$\frac{K(K+1)}{2} > n$

$K^2 > n$

$K > \sqrt{n}$

$O(\sqrt{n})$.

Ans-2

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

$T(0) = T(1) = 1$

if $T(n-1) \approx T(n-2)$

(Lower Bound) $T(n) = 2T(n-2)$

$= 2\{2T(n-2)\}$

$= 4T(n-4)$

$= 4(2T(n-6))$

$= 8T(n-6)$

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

$= 16T(n-8)$

$\vdots$

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

$n-2k=0$

$n=2k$

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

$= 2^{n/2}$

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

if $T(n-2) \approx T(n-1)$

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

$n-k=0$

$k=n$

$$T(n) = 2^k \times T(0) = 2^n$$

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

Ans-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=1; i<=n; i=i*2)
{
    for(int j=1; j<=n; j=j*2)
    {
        //some O(1)
    }
}
```

Ans-4

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

Let assume

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

so

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

applying masters.

$$n^{\log_b^a} = n^{\log_2^2} = n$$

$$f(n) > n$$

$$\text{So} \quad T(n) = \Theta(n^2).$$

Ans-5.

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

$i=1$ — $j=1$, $j=2$, $j=3$ → n times.
⋮
$j=n$

$i=2$ — $j=1$, $j=3$, $j=5$ ⋮

loop ends when $j>n$

$1+3+5+7 > n$

$K = n/2$

– n times

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

$= O(n^2)$

Ans-6 for(int i=2; i<=n; i=Pow(i,k))

{

//some(1)

}

Complexity of Pow(i,k) – O(logN)

= log(k)

$i=2$

$i=2^k$

$i=2^{k^2}$

$i=2^{k^3}$

⋮

$i=2^{k^n}$

Loop ends when $i>n$

$$2^{k^m}>n$$

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

$$k^m > \log n$$

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

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

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

Ans-8 a) $100 < \log n < \sqrt{n} < n < \log(\log n) < n\log n < \log n! < n! < n^2 < \log 2^n < 2^n < 2^{2n} < 4^n$.

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

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