Tutorial - 02

Q. 1 Solution

i	j
0	1
1	2
3	3
6	4
10	5
⋮	⋮

No. of Times loop is running by i

$S_k = 1 + 3 + 6 + 10 + \_\_\_\_ + T_k$

$S_{k-1} = 1 + 3 + 6 + \_\_\_\_ T_{k-1}$

Substracting both

$S_k - S_{k-1} = 1 + 2 + 3 + 4 + \dots + (k-1)$

$T_k = \frac{(k-1)k}{2}$

Given that k^{th} term is n.

$T_k = n$.

$\frac{k(k-1)}{2} = n \Rightarrow \frac{k^2}{2} - \frac{k}{2} = n$

$= k^2 = n$

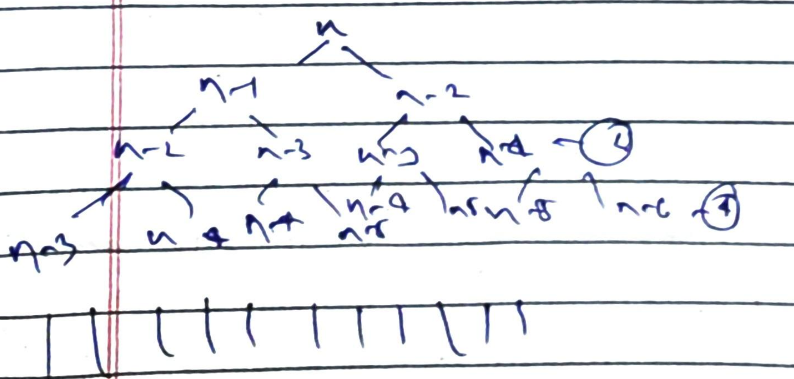
$\sqrt{n} = \sqrt{k}$

$= T(n) = O(\sqrt{n})$

$= \boxed{T(n) = O(n^{1/2})}$ Answer

Q.2. Solution

$T(n) = T(n-1) + T(n-2) + O(1)$ for recursive fibonacci.



$s = 1 + 2 + 4 + \_\_\_\_ + 2^n$

$= \frac{2^{n+1} - 1}{2 - 1} = 2^{n+2} - 1.$

Time Complexity
$T(n) = O(2^n)$.

$\Rightarrow$ Space Complexity
$= O(n)$

PAGE NO.:
DATE: / /

Solution 3

Code having time complexity :

O(n logn) =
```
for (int i=1 ; i<=n ; i++ )
{
    for (int j=1 ; j<n ; j=j*2)
    {
        printf("Hello");
    }
}
```

O(n³) =
```
for (int i=0; i<n; i++ )
{
    for (int j=0; j<n ; j++ )
    {
        for (int k=0; k<n; k++ )
        {
            printf("Hello") ;
        }
    }
}
```

O(log(logn)) =
```
for (int i=2 ; i<=n; i=pow(i,3))
{
    printf("Hello");
}
```

* ~~Hello~~
Here n can be any positive number.

PAGE NO.
DATE

Q.4 Solution

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

Ignoring lower order terms.

$$T(n) = T\left(\frac{n}{2}\right) + cn^2.$$

Using Master Theorem

$a = 1,\ b = 2,\ f(n) = n^2.$

$c = \log_b a = \log_a 1 = 0.$

$\boxed{0 < n^2}$ True

$\Rightarrow \boxed{T(n) = \Theta(n^2)}$

Q.5 Solution

i	j
1	n
2	n/2
3	n/3
4	n/4
⋮	⋮

Time Complexity will be sum of series.

$$S = \frac{n}{1} + \frac{n}{2} + \frac{n}{2} + \ldots$$

$$= \sum_{i=1}^{n} \left(\frac{n}{i}\right)$$

$$\text{Complexity} = n \times \sum_{i=1}^{n} \left(\frac{n}{i}\right)$$

$$\boxed{T(n) = n\log n}$$

Q.6 Solution

Sequence.

$2, 2^k, (2^k)^k, ((2^k)^k)^k \ldots$

Generalizing

$= 2^{k^0}, 2^{k^1}, 2^{k^2}, \ldots \ldots 2^{k^{d-1}}$

Assumption

Given : last term is n.

$\Rightarrow 2^{k^{x-1}} = n$

$k^{x-1} \log 2 = \log n$

$(x-1) \log k = \log (\log n)$

$x = \log (\log n)$

Time Complexity : $\boxed{T(n) = O(\log(\log n))}$

Question 8.

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

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

(c) $96 < \log_e n < \log_2 n < 5n < n(\log_2 n) < n(\log_2 n) < \log(n!) < 8n^2 < 7n^3 < n! < 8^{2n}$.