Tutorial-2

Ques 1. Let's take $n = 4$,

① → $i = 1$ $s = 1$

$s <= n$ ⇒ $1 <= 4$

$i = 2$ ⇒ $s = 2+1 = 3$

② → $s \leq n$ ⇒ $3 <= 4$

$i = 3$ $s = 3+3 = 6$

The s here Increases with Rate of i

$\therefore 1 + 2 + 3 + \ldots + K$ [K = Iteration for which $s <= n$]

⇒ $\frac{K(K+1)}{2} < n$

⇒ $\frac{K^2+K}{2} < n$ ⇒ $\boxed{K^2 = O(\sqrt{n})}$

Ques 2. for fibonacci Series Recurrence Relation

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

$\left[\begin{matrix} T(1) = 1 \\ T(0) = 0 \end{matrix}\right]$

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

$= T(1) + 0 + 1 = 2$

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

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

$TC = 2^0 + 2^1 + 2^2 + \ldots 2^n$

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

$\boxed{TC = O(2^n)}$

Since the call stack value never rises above n So,

$\boxed{SC = O(n)}$

Ques 3 *
```
for(int i=0 ; i<n ; i++){
    for (int j=n; j>0; j=j/2){
        cout << "*" << endl;
    }
}
```

TC = O(n log n)

*
```
for(int i=0 ; i<n ; ++i)
    for (int j=0 ; j<n; ++j)
        for( int k=0 ; k<n; ++k)
            cout << "*" << endl;
```

TC = O(n^3)

*
```
for( int i=n ; i>0 ; i=i/2)
    for( int j=n ; j>0; j=j/2)
        cout << "*" << endl;
```

TC = O(log(log n))

Ques 5 Inner loop executes n/i times for each value of i. Its running time is $O(n \log n)$

Ques 6. In this case i takes values $2, 2^k, 2^{k^2}, 2^{k^3} \ldots$

$\ldots 2^{\log_k(\log(n))}$

The last term must be less than or Equal to n and we have $2^{k \log_k(\log(n))} = 2^{\log(n)} = n$

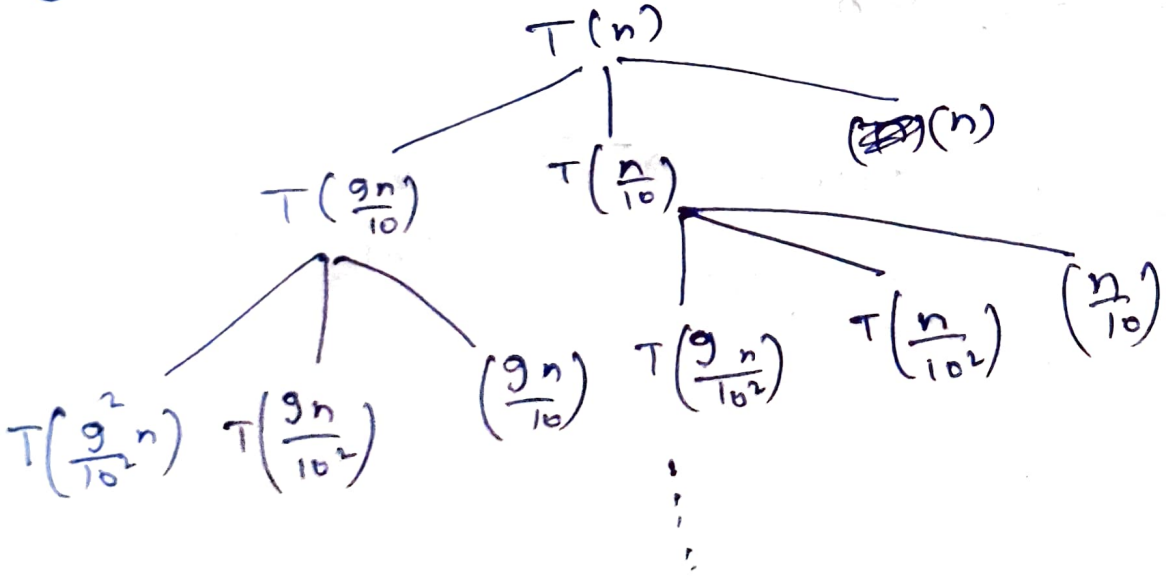
So there are in total $(\log_k(\log n))$ many iterations

$\therefore TC = O(\log(\log(n)))$

Ques 7. If we Split in this Manner then the recurrence Relation will be

$T(n) = T(9n/10) + T(n/10) + O(n)$

where the first branch of the size $9n/10$ and second one is $n/10$ so branches have nodes in 9:1 ratio



At 1st level the cost = n

At 2nd level the cost = $\frac{9n}{10} + \frac{n}{10} = n$

∴ Time Complexity = Summation of cost of all levels

$= O(n \log_{10/9} n)$ for longer branch

$= \Omega(n \log_{10} n)$ for shorter branch

The base of log does not matter as it is only a matter of constant.

Ques 8 (a) $2^{2^n} > n! > 4^n > 2^n > n^2 > n\log n > \log(n!) > n > \log^2 n > \log n > \sqrt{n} > \log(\log n) > 100$

decreasing Rate of Growth.

(b) $n! > 2(2^n) > n\log(n) > \log(n!)$

$> 4n > 2n > 2\log(n) > \log 2n$

$> \log(n) > \sqrt{(\log n)} > \log(\log(n)) > 1$

(c) $8^{2^n} > n! > 7n^3 > 8n^2 > n\log_6(n) > n\log_{42}(n)$

$> \log(n!) > \log_8(n) > \log_2(n) > 5n > 96$