Tutorial-2

Ques-1) What is the time Complexity of below code & how ?

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

$j=1 \quad i=1$
$j=2 \quad i=1+2$
$j=3 \quad i=1+2+3$
} m-level

for(j)

$\therefore 1+2+3+\dots < n$

$1+2+3+m < n$

$\frac{m(m+1)}{2} < n$

$m = \sqrt{n}$

By Summation method

$\sum_{j=1}^{m} 1 \Rightarrow 1+1+\dots+\sqrt{n}$ times

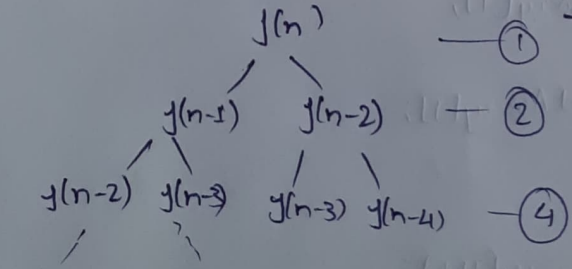
$\boxed{T(n) = \sqrt{n}}$ Ans.

Ques 2) What recurrence relation for function that prints Fib. Series. Solve it do get the time Complexity. What will be space Complexity & why

For Fibonacci Series

$f(n) = f(n-1) + f(n-2)$ $\quad f(0)=0$, $f(1)=1$

By forming a tree,



$1, 2, 4, 8 \dots n$

$a = 1, \; r = 2$

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

$\boxed{T.C = O(2^n)}$

Space Complexity

Recursive :- $T(n) = O(n)$

Iterative :- $T(n) = O(1)$

Ques. 3) Write programs which have time Complexity $n\log n$, n^3, $\log(\log n)$

① $n\log n$ → MergeSort

```
Void mergeSort(int array[], const int low, const int high)
{
  if(low>=high) return;
  int mid = low+(high-low)/2;
  mergeSort(array, low, mid)
  mergeSort(array, mid+1, high)
  merge(array, low, mid, high)
}

Void merge(int array[], const int low, int mid, int high)
{
  int i, j, k;
  int n1 = mid-low+1;
  int n2 = high-mid;
  int leftArray[n1], rightArray[n2]
  for(int i=0; i<n1; i++)
    leftArray[i] = array[low+i];
  for(int j=0; j<n2; j++)
    rightArray[i] = Array[i+mid+1];
  i=0, j=0, k=low;
  while(i<n1 && j<n2)
  {
    if(leftArray[i] <= rightArray[j])
        array[k] = leftArray[i];
    else
        array[k] = rightArray[j];
  }
  while(i<n1)
    array[k++] = leftArray[i++];
  while(j<n2)
    array[k++] = rightArray[j++];
}
```

(ii) n^3 → Multiplication of two matrices.

```
for(i=0; i<n1; i++)
for(j=0; j<n2; j++)
  for(k=0; k<n1; k++)
  {
    res[i][j] += a[i][k] * b[k][j];
  }.
```

(iii) log(log(n))-

```
for(i=2; i<n; i=i*i)
  count++;
```

Ques-4) Solve the following recurrence relation:-

At level

$0 \rightarrow cn^2$

$1 \rightarrow \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{c5n^2}{16}$

$2 \rightarrow \frac{n^2}{8^2} + \frac{n^2}{16^2} + \frac{n^2}{4^2} + \frac{n^2}{8^2} = \left(\frac{5}{16}\right)^2 n^2 c2$

!

$\text{max level} = \frac{n}{2^k} = 1$

$= k = \log_2 n$

$T(n) = c\left(n^2 + \frac{5}{16}n^2 + \frac{5^2}{16}n^2 + \cdots \left(\frac{5}{16}\right)^{\log n} n^2\right)$

$T(n) = cn^2\left[1 + \frac{5}{16} + \frac{5^2}{16} + \cdots + \frac{5}{16}\log n\right]$

$T(n) = cn^2 \times 1 \times \left(\frac{1 - (5/16)\log n}{1 - (5/16)}\right)$

$T(n) = cn^2 \times \frac{11}{5} \times \left(1 - \left(\frac{5}{16}\right)^{\log n}\right)$

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

$\boxed{O(cn^2)}$ Ans.

Ques-5) what is time complexity of following func()?

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

for	i	j	j=(n-1) i times
	1	1	
	2	1+3+5	
	3	1+4+7	
	⋮		
	n		

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

$$T(n) = \frac{n-1}{1} + \frac{(n-1)}{2} + \frac{(n-1)}{3} + \cdots \frac{n+1}{n}$$

$$T(n) = n\left[1+\frac{1}{2}+\frac{1}{3}+\cdots\frac{1}{n}\right] - 1\left[1+\frac{1}{2}+\frac{1}{3}+\cdots\frac{1}{n}\right]$$

$$= n\log n - \log n$$

$$\boxed{T(n) = O(n\log n)}$$ Ans.

Ques-6) what should be time complexity of

```
for(int i=2; i<=n; i=pow(i,K))   K -> Constant
  // Some O(1)
for
```

for	i
	2^1
	2^K
	2^{K^2}
	2^{K^3}
	⋮
	2^{K^m}

$\therefore \sum_{i=1}^{m} 1$

where

$2^{K^m} <= n$

$K^m = \log_2 n$

$m = \log_K \log_2 n$

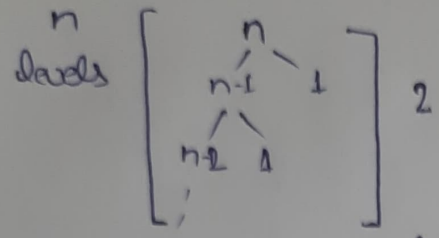
$1+1+1 \cdots$ m times

$T(n) = O(\log_K \log n)$ — Ans

Ques. 7)

Sol:- Given Algorithm divides array in 99% and 1% part

$\therefore T(n) = T(n-1) + O(1)$



'n' work is done at each level

$T(n) = (T(n-1) + T(n-2) + \dots T(1) + O(1)) \times n$

$= n \times n$

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

Lowest height = 2

highest height = n

$\boxed{\therefore \text{ different} = n-2}$ $n > 1$

The given algorithm produces linear result.

Ques 8) Arrange the following in increasing order of growth:

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

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

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

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

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

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