Tutorial-2

Q: what is time complexity of below code.

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

$j=1$ $\quad$ $i=1$

$j=2$ $\quad$ $i=1+2$

$j=3$ $\quad$ $i=1+2+3$

for (i)

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

$\therefore$ $1+2+3+ \cdots m < n$

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

$m \simeq \sqrt{n}$

By summative method

$\sum_{i=1}^{m} 1 \Rightarrow 1+1+ \cdots \sqrt{n}$

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

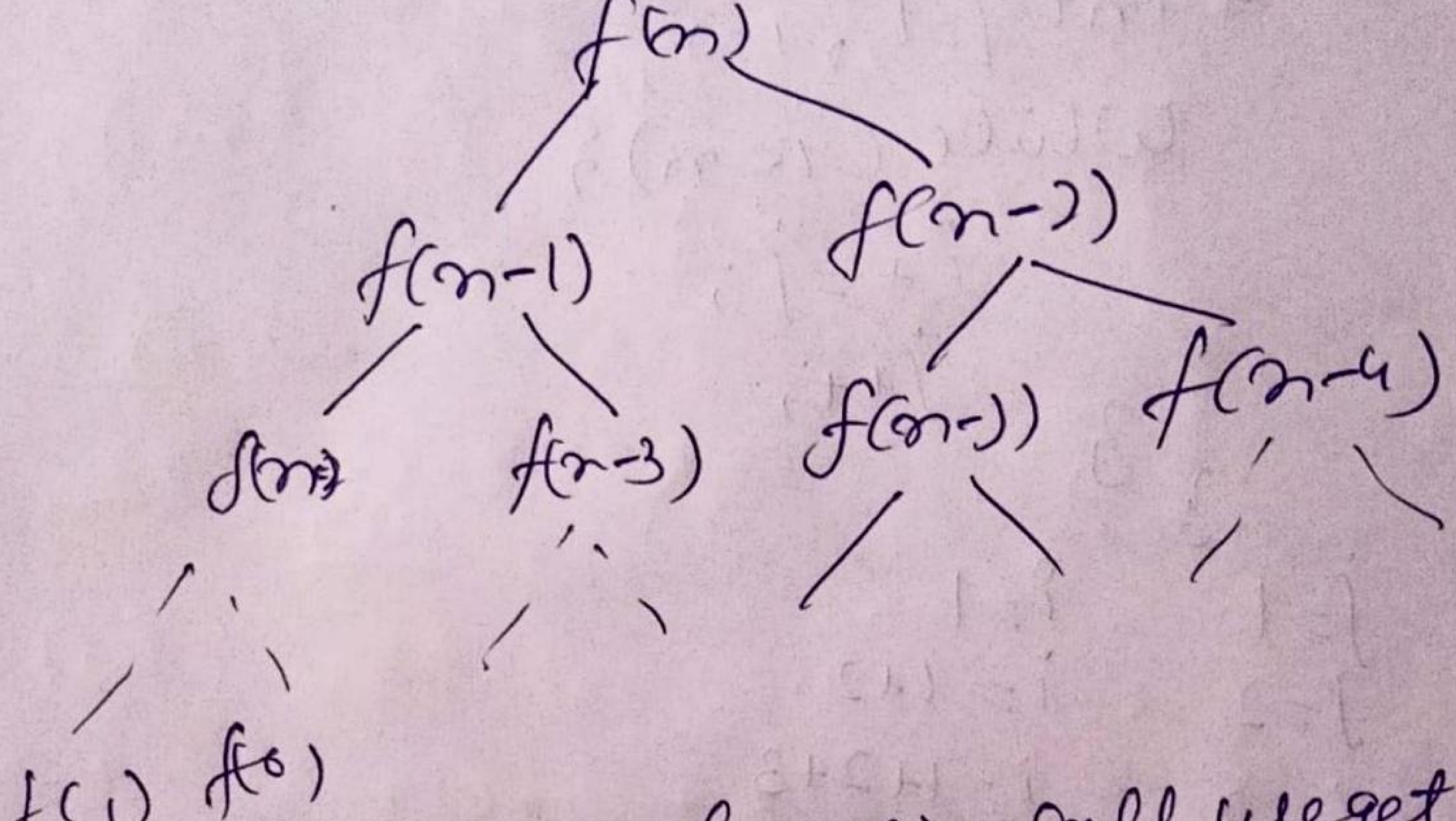
Q

→ For Fibonacci Series;

$f(n) = f(n-1) + f(n-2)$ $f(0)=0$

$f(1)=1$

By forming a tree



∴ At every function call we get 2 fn calls

∴ for n levels we have $= 2 \times 2 \cdots n$ times

∴ $\boxed{T(n) = 2^n}$

Maximum Space: Considering Recursive Stack: no of maxi calls $= n$

for each call we have space complexity $O(1)$

∴ $[T(n) = O(n)]$

without recursive stack: each call $[T(n) = O(1)]$

Q: Write program which have complexity;

n(logn) , n^3, log(logn)

1) n log(n) → Quick Sort

```
void quicksort (int arr[],
                 int low, int high)
{
  if (low<high)
  { int pi= partition(arr, low,high);
        quicksort (arr, low, pi-1);
        quick sort(arr, pi+1, high);
  }
}
int partition (int arr[], int low, int
                          high)
{
    int pivot= arr[high];
      int i=(low-1);
   for (int j=low; j<=high-1; j++)
      { if (arr[i] <pivot)
         { i++;
           swap (&arr[i], &arr[j]);
         }
      } swap (&arr[i+1], &arr(high])
         return i+1;
}
```

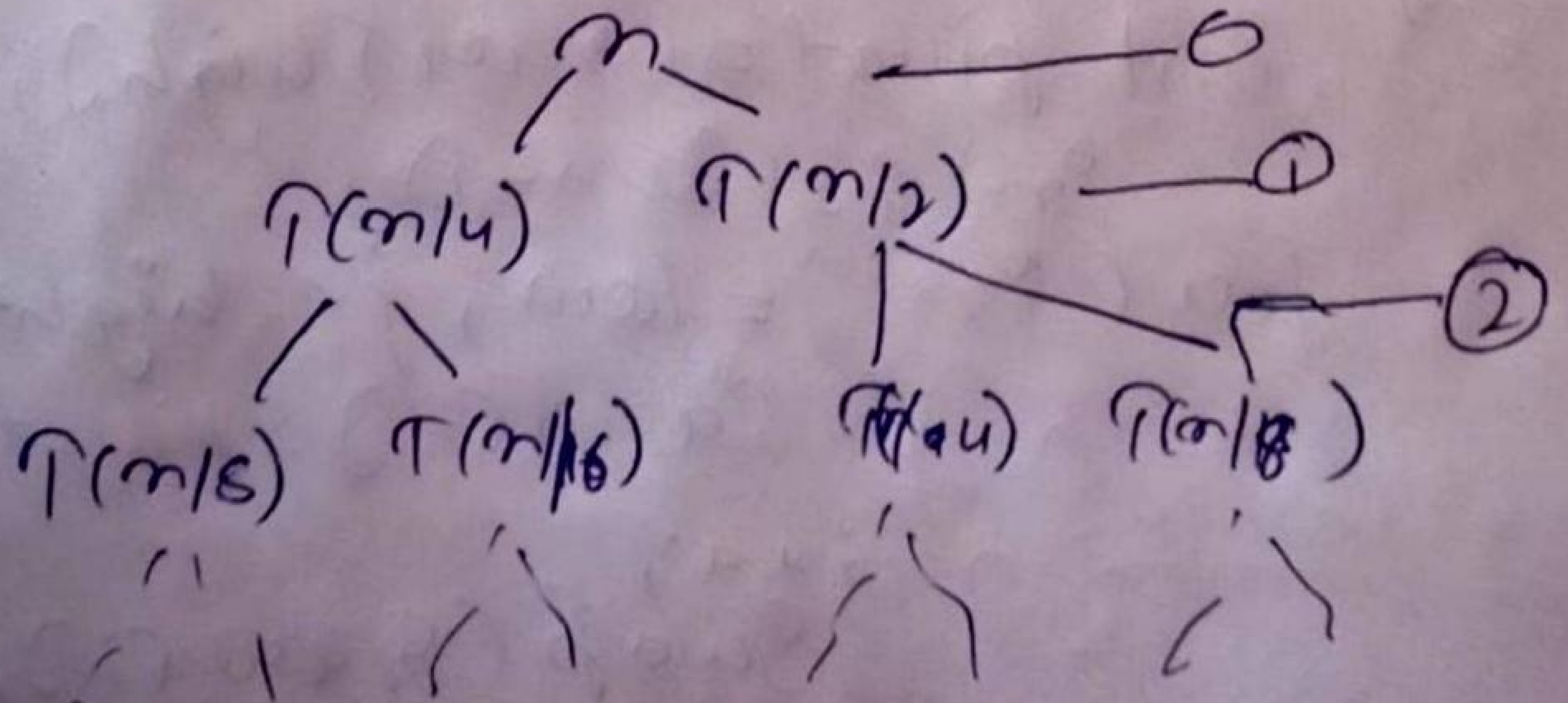
2) n^3 Multiplication of 2 square matrix :

```
for ( i=0; i < n' ; i++)
  for ( j=0 ; j<2 ; j++)
   for (k=0 ; k<k1 ; k++)
   {
     res[i][j]+= a[i][k] *
                 b[k][j];
   }
```

3) log(log(n))

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

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



at level

0 → cn^2

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

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

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

$= k = \log_2 n$

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

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

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

$T(n) = cn^2 + \frac{11}{3} \times \left(1 - \left(\frac{5}{16}\right)\log n\right)$

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

$\underline{O(n^2)}$

Q: Time complexity of

```
for (int i=2; i<n; i=pow(i,k))
{   // some O(1)
}
```

where k is a const^n

for i — where

2^1 — $2^{k^m} <= n$

2^k — $k^m = \log_2 n$

2^{k^2} — $m = \log k \log_2 n$

2^{k^3}

$\vdots$

2^{k^n}

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

$1 + 1 - -$ n times

$T(n) = O(\log_k \log n)$ ans

Q: Arrange following in increasing order.

a) $n, \log n, \log\log n,$ $\log(n!), n\log n,$

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

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

$< n < n\log n < \log(n!) < n^2 < 2^n$

$< 4^n < 2^{2^n}.$

b) $2(2^n), 4n, 2n, 1, \log(n), \log(\log(n)),$

$\sqrt{\log(n)}), \log 2(n), 2\log(n), \log(n!),$

$n!, n^2, n\log(n).$

$\rightarrow 1 < \log\log(n) < \sqrt{\log n} < \log n < \log 2n <$

$2\log n < n < n\log n < 2n < 4n < \log(n!)$

$< n^2 < n! < 2^{2n}.$

$\rightarrow 96 < \log n < \log 2n < 5n < n\log_6(n) <$

$n\log_2(n) < \log(n!) < 8n^2 < 7n3$

$< n! < 8^{2n}$