Satyam Gupta
CST
32

PAGE _____
DATE _____

TUTORIAL-2

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

for
$j=1$	$i=1;$	m levels
$j=2$	$i=1+2;$	
$j=3$	$i=1+2+3;$	

for (i)

$\because$ $1+2+3+\dots \alpha n$

$\because$ $1+2+3+m \alpha n$

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

$m \approx \sqrt{n}$

$\because$ by summation method

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

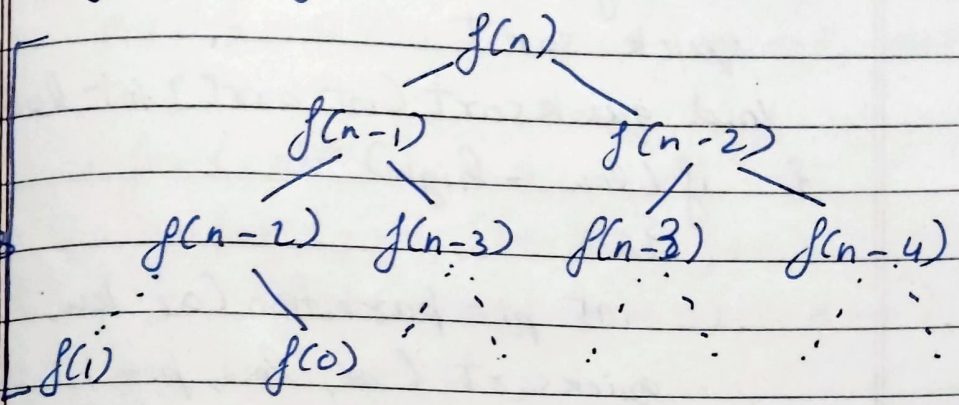
$\because \boxed{T(n) = \sqrt{n}}$

Q-2 For Fibonacci series:-

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

$f(0) = 0 \quad f(1) = 1$

By forming tree :-



∴ At every function call we get 2 fⁿ calls.
∴ for n levels :-
we have = 2 × 2 ... n times.
∴ $\boxed{T(n) = 2^n}$

Maximum space :-
considering recursive stack :
no. of cells maximumum = n

for each call we have space complexity
O(1)
∴ T(n) = O(n)
without considering recursive stack
or
Each call we have time complexity O(1)
∴ (T(n) = O(1))

8-3 ① n log n

quick sort

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

② n^3

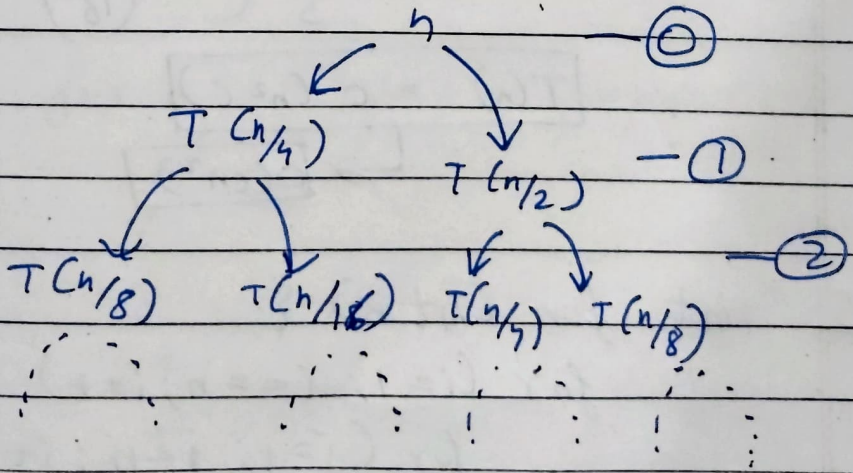
multiplication of two square matrix

```
for (i=0; i<r1; i++)
    for (j=0; j<c2; j++)
        for (k=0; k<c1; k++)
        {
        res[i][j]+=a[i][k]*b[k][j];
        }
```

③ log(log n)

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

Q-4 $T(n) = T(n/4) + T(n/2) + c \times n^2$



At level :-

0 $\longrightarrow cn^2$

1 $\longrightarrow \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{5n^2}{16}$

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

⋮

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

$\Rightarrow k = \log_2 n$

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

$= \left(\left(\frac{5}{16}\right)^{\log N} n^2\right)$

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

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

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

$\therefore \boxed{T(n) = O(n^2 c)}$

$\hookrightarrow \boxed{O(cn^2)}$

Q-5

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

for i	j	$f = (n-1)/i$ times
1	1	
2	1+3+5	
3	1+4+7	
⋮	1+5+9	
n	⋮	

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

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

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

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

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

Q-6) for (i = 2; i <= n; i = pow(i, k))

O(1)

i
2^1
2^k
2^{k^2}
2^{k^3}
⋮
2^{k^m}

where, $2^{k^m} = 2 = n$

$k^m = \log_2 n$

$m = \log_k \log_2 n$

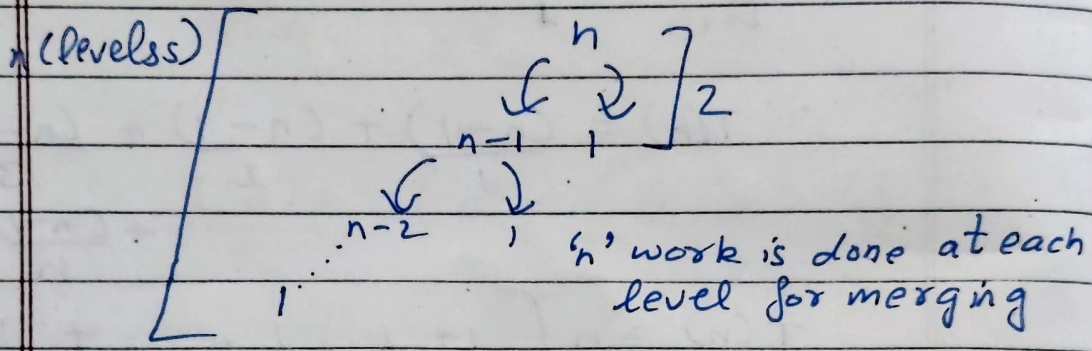
$\because \sum_{i=1}^{m} 1$

$\Rightarrow 1 + 1 + 1 \ldots$ m times

$\Rightarrow \boxed{T(n) = O(\log_x \log n)}$

8-7 Given algo divides array in 99% & 1% part (sating algo)

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



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

$= n \times n$

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

Lowest height = 2

heighest height = n

$\therefore \boxed{\text{diff.} = n-2} \quad n > 1$

The given algo provides linear result.

Q.8 Considering for large values of 'n'

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) $96 < \log_8 n < \log 2n < 5n < n\log_8 n < n\log_6 n < \log(n!) < 8n^2 < 7n^3 < n! < 8^{2n}$.