Date.
Page No.

Tutorial - 2

Name - Shagun Gupta
Section - C
Roll No. - 11
University RollNo. - 2017012

Q1.

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

for (i)

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

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

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

$m \simeq \sqrt{n}$

By summative method

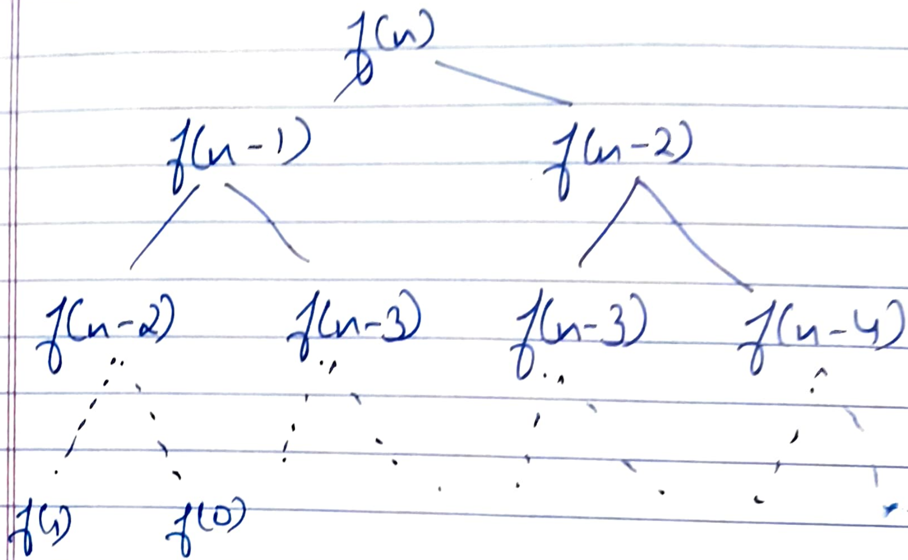
$\sum_{i=1}^{m} 1 \Rightarrow 1+1+\dots+\sqrt{n}$ times

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

Q2. 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 f^n calls

∴ for n levels.

we have = 2 × 2 .... n times

∴ $T(n) = 2^n$

Maximum Space

considering Recursive Stack:

no. of maximum calls = n

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

∴ $T(n) = O(n)$

without considering recursive stack each call we have time complexity $O(1)$

∴ $T(n) = O(1)$

Date.
Page No.

Q3

1

nlogn → QuickSort

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

2. n^3 → multiplication of 2 square matrix.

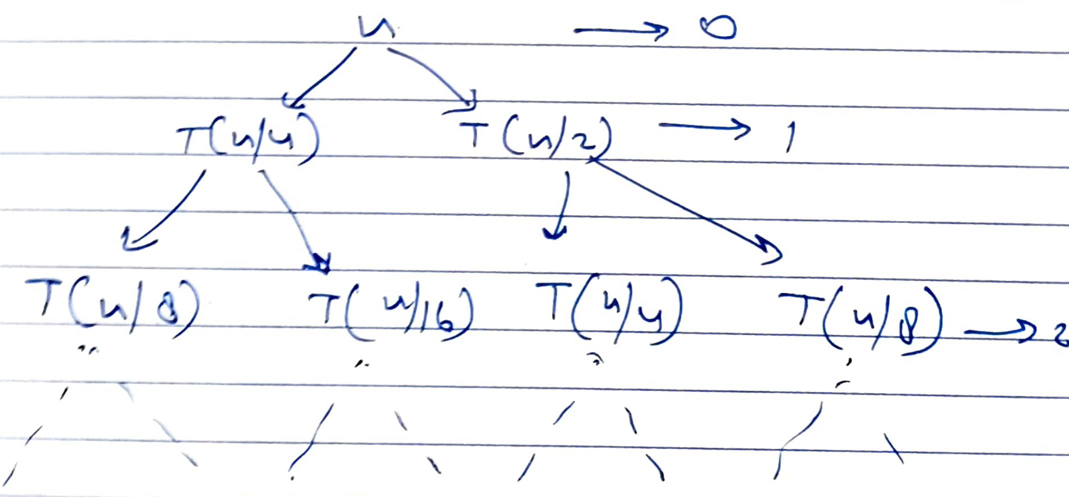
```
for (i=0; i<r1; i++).
    for (j=0; j<c2; j++
```

```
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)
{
    count ++;
}
```

Q4



at level

$0 \rightarrow Cn^2$

$1 \rightarrow \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{C5n^2}{16}$

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

⋮

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

$= k = \log_2 n$

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

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

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

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

$$T(n) = O(n^2(1)$$
$$= O((n^2))$$

Q5 for

i	j
1	1
2	1+ 3+5
3	1+4+7
⋮	
n	

$j = (n-1)1$ times

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

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

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

Date.

Page No.

$= n\log n - \log n$

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

Q6.

for i where

2^1

2^k $\quad 2^{k^m} <= n$

2^{k^2} $\quad k^m = \log_2 n$

2^{k^3} $\quad m = \log_k \log_2 n$

$\vdots$

2^{k^n}

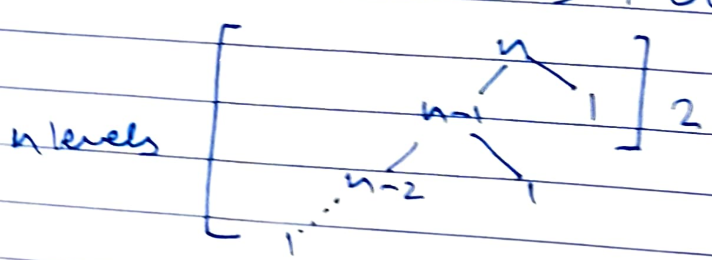
$\therefore \sum_{i=1}^{m} 1$

$1 + 1 + 1 + 1 + \dots$ m times

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

Q7 Given algo divides array in 99% and 1% part.

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



work is done at each level

$T(n) = [T(n-1) + T(n-2) + \dots + T(1) + d[1]) \times n$

$= n \times n$

$\therefore T(n) = O(n^2)$

lowest height $= 2$

highest height $= n$

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

The given algorithm produces linear result

Q8. a) $100 < \log\log n < \log n < (\log n)^2 < \sqrt{n} < n < n\log n < \log(n!) < n^2 < 2^n < 4^n < 2^{2n}$

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

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