TUTORIAL-2

Ques 1

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

for $j=1$ $\quad$ $i=1$;

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

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

$\vdots$ $\quad$ $\vdots$

(m levels)

for i

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

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

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

$m \approx \sqrt{n}$

$\therefore$ by Summation method

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

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

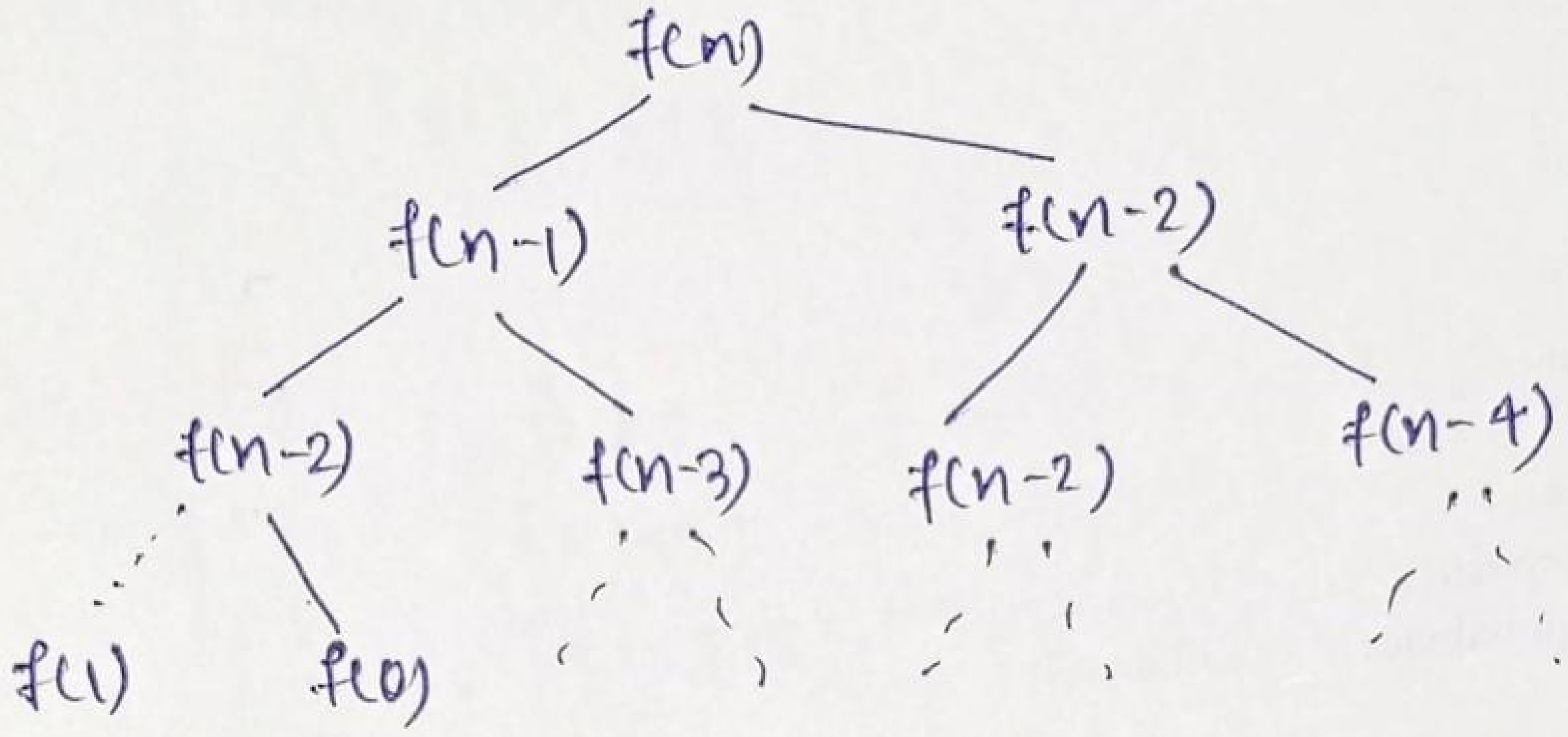
Scanned by TapScanner

Que2. 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 function calls -

$\therefore$ For n levels -

we have, $2 \times 2 \times \ldots n$ times

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

Maximum space -

considering recursive stack,

no. of calls maximum $= n$

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

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

without considering recursive stack,

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

$\therefore \boxed{T(n) = O(1)}$

Ques 3.

(i) n.logn

Quick sort

```
void quicksort (int arr[], int low, int high)
{
    if (low < high)
    {
        int pi = ~~posit~~ partition (ar, low, high);
        quicksort (ar, low, pi-1);
        quicksort (ar, pi+1, high);
    }
}

int partition (int ar[], 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

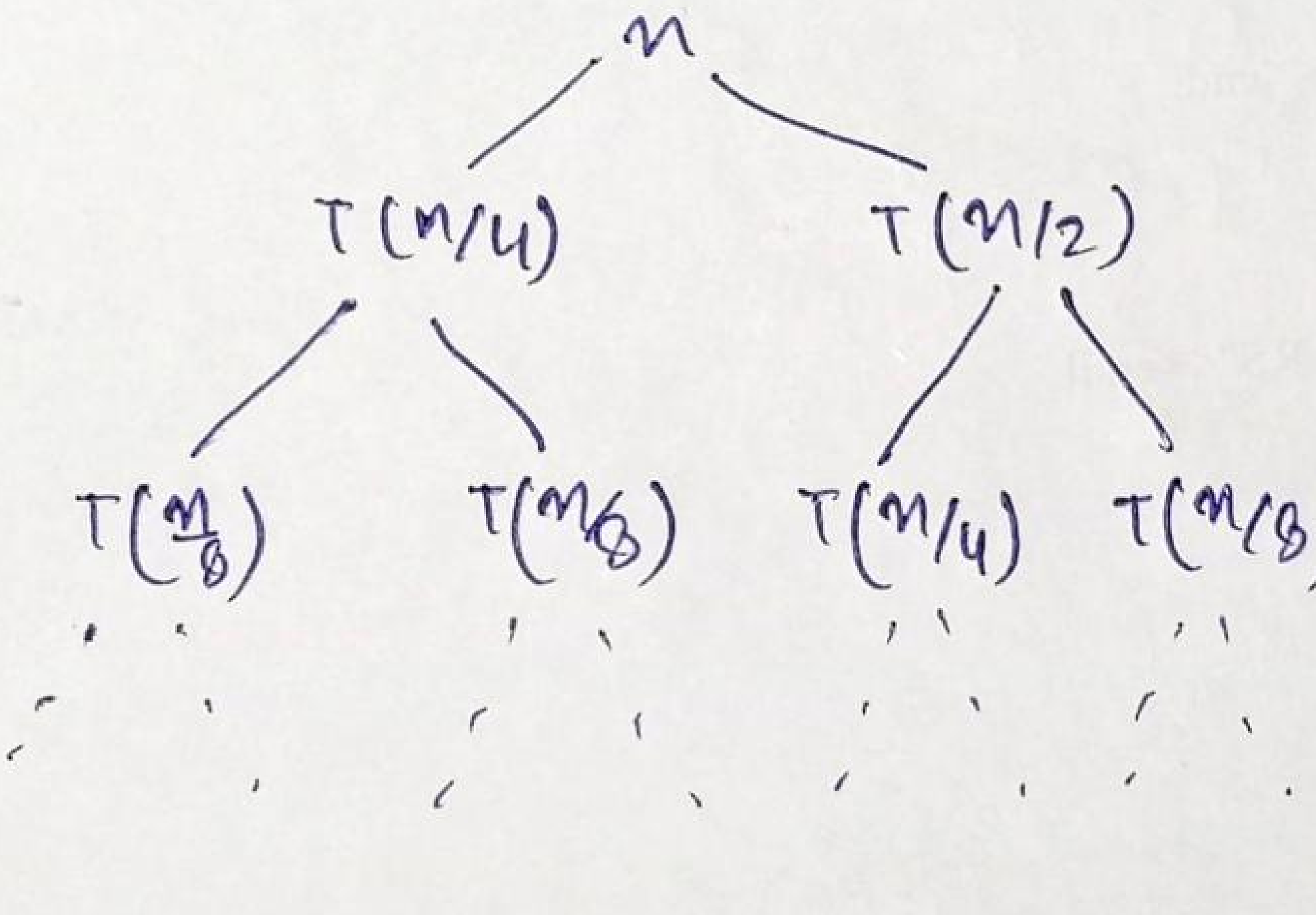
Multiplication of two square matrix

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

<u>Que 4</u>: $T(n) = T(n/4) + T(n/2) + c \times n^2$



<u>At level -</u>

$0 \rightarrow cn^2$

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

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

⋮

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

$$\Rightarrow x = \log_2 n$$

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

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

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

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

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

Ques:

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

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

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

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

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

$$= n\log n - \cancel{\log n}\log n$$

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

Que6. for (i=2; i<=n; i = pow(i,k))
{
    O(1)
}

for i

2^1

2^k

2^{k^2}

2^{k^3}

⋮

2^{k^m}

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

$k^m = \log_2 n$

$m = \log_k \log_2 n$

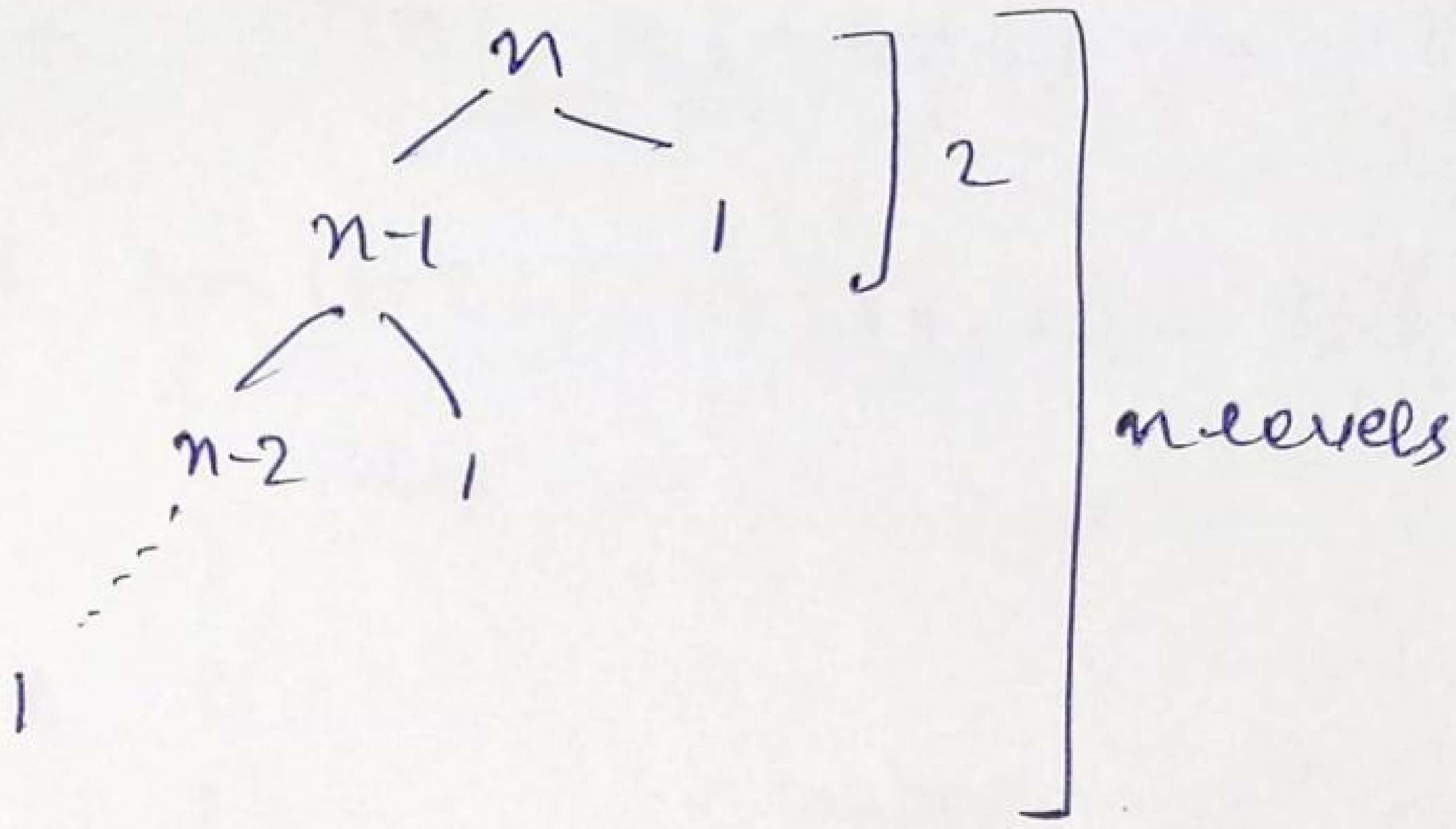
$$\therefore \sum_{i=1}^{m} 1$$

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

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

Que 7. Given algo divides array in 99% & 1% part

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



'n' work is done at each level for merging.

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

$= n \times n$

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

Lowest height = 2

Highest height = n

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

The given algo produces linear result.

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