Solution1

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

$\left.\begin{array}{ll} j=2 & i=1+2 \\ j=3 & i=1+2+3 \end{array}\right\}$ m-level

for (i)

$\because 1+2+3+\ldots+<n$

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

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

$m \approx \sqrt{m}$

By summation method

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

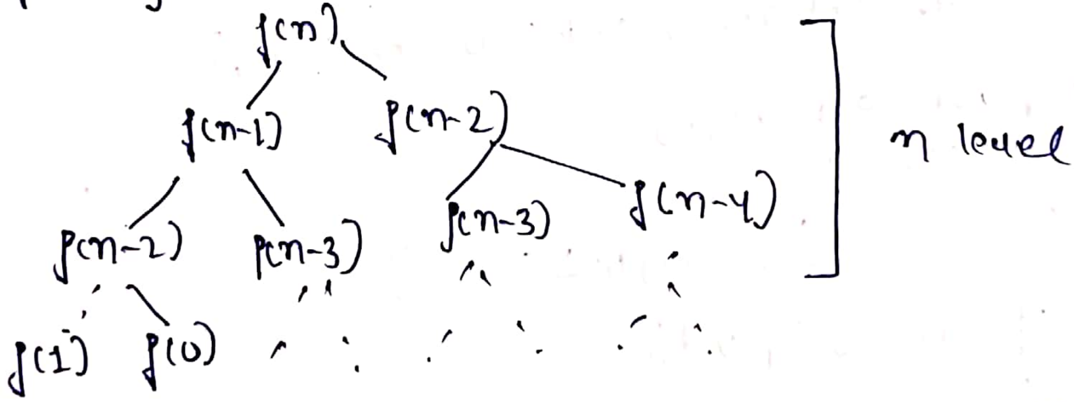
$\boxed{T(n)=\sqrt{n}}$

Question 2

In Fibonacci series $f(0)=0, f(1)=1$

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

By forming tree



$\because$ At every function call we get 2 function calls

$\therefore$ for n level

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

$\because$ $T(n) = 2^n$

Maximum space:

considering Recursive stack:

no. of calls maximum $= n$

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

$\because$ $T(n) = O(n)$

without considering Recursion stack:

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

$\because$ $T(n) = O(1)$

Ques 3

1) nlogn → Quick sort

```
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[i] < pivot)
    { i++;
    swap [& arr[i], & arr[j]]);
  }
}
```

```
swap (& arr[i+1], & arr[high]);
  return (i+1);
}
```

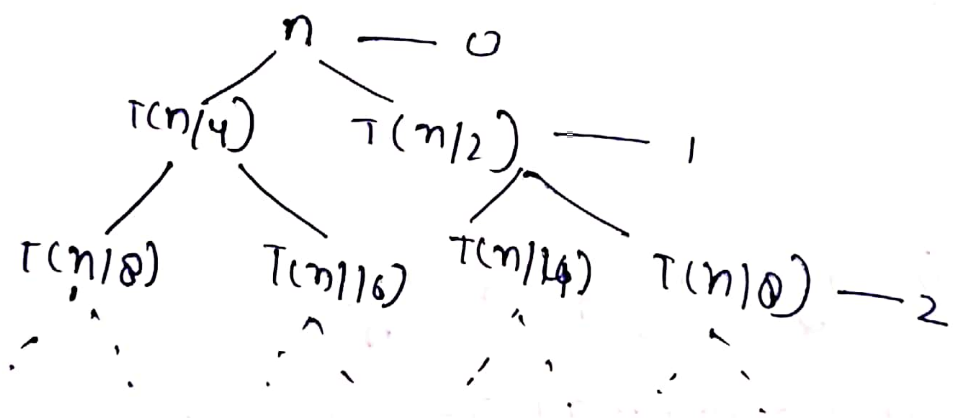
② n^3 → Multiplication of 2 square matrix

```
for(i=0; i<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++;
  }
```

ques 4



At level

$0 \rightarrow cn^2$

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

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

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

$\Rightarrow k = \log_2 n$

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

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

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

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

$O(cn^2)$ Ans

Question 5

i	j
1	1
2	1+3+5
3	1+4+7
⋮	⋮
n	

$j = (n-1)/i$ times

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

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

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

$= n\log n - \log n$

$T(n) = O(n \log n) \rightarrow$ Ans

Question 1

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

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

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

Sol 6

i

2^1

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

2^{K^2} $K^m = \log_2 n$

2^{K^3} $m = \log_K \log_2 m$

$\vdots$

2^{K^m}

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

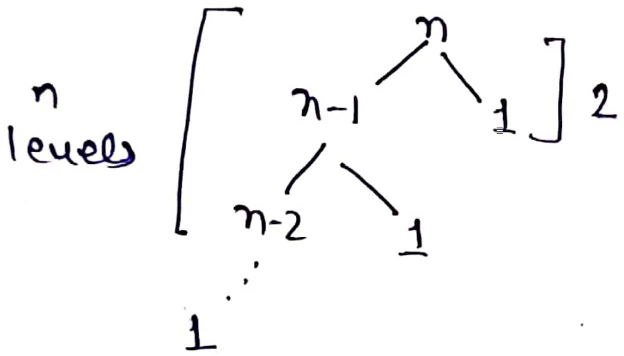
$1 + 1 + 1 \ldots m$ times

$T(n) = O(\log_K \log n) \rightarrow$ Ans

Ques 7

Given algorithm divides array in 99% and 1% part

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



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

$= n \times n$

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

lowest height $= 2$

heighest height $= n$

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

the given algorithm produces linear result