Tutorial - 2

Name → Tanvi Nautiyal
Section → G
Roll No. → 47
University Roll No. → 2017090

Q.1 What is the time complexity of below code & how?

```
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)

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

$1 + 2 + 3 + m < n$

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

$m \simeq \sqrt{n}$

By summation method

$\frac{m}{i+1} \; 1 = 1 + 1 + \ldots \sqrt{n}$ times

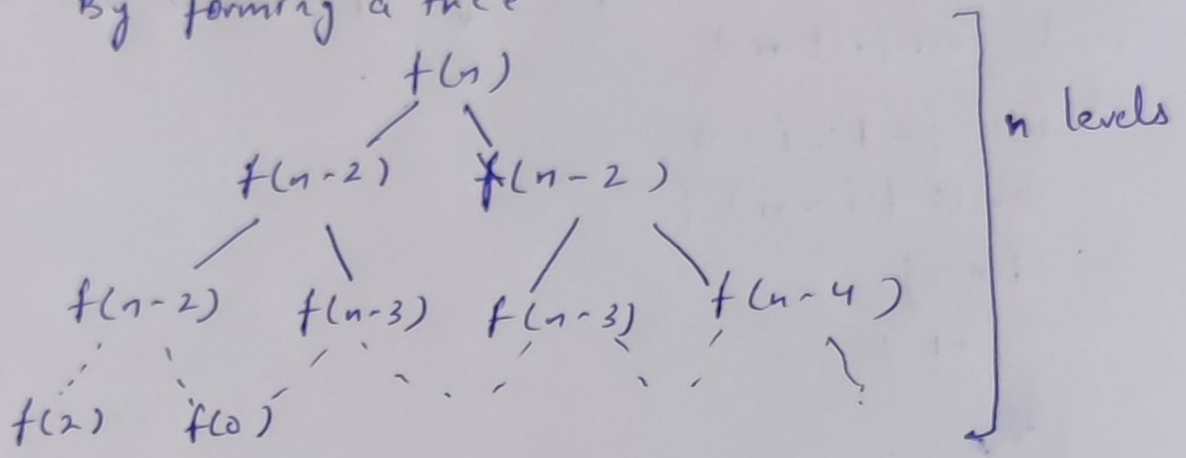
$T(n) = \sqrt{n}$ Ans.

Q.2 Write reccurance relation for function that prints Fibonacci series. Solve it to get the time complexity. What will be the space complexity and why?

→ For Fibonacci series

$f(n) = f(n-1) + f(n-2)$ $\quad f(n)=0$, $f(1)=1$

By forming a tree



∴ In every function call we get 2 function calls

∴ for n levels

We have = 2 x 2 . . . n times

$\underline{T(n) = 2^n}$

Maximum space:

Considering Recursion

Stack:

No. of calls maximum = n

For each call we have space complexity O(1)

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

Without considering Recursion stack:

Each call time complexity O(1)

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

Q.3. Write programs which have complexity : $n(\log n)$, n^3, $\log(\log n)$

1) $n \log n$ → Quick sort

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

2) n^3 → Multiplication of 2 square matrix

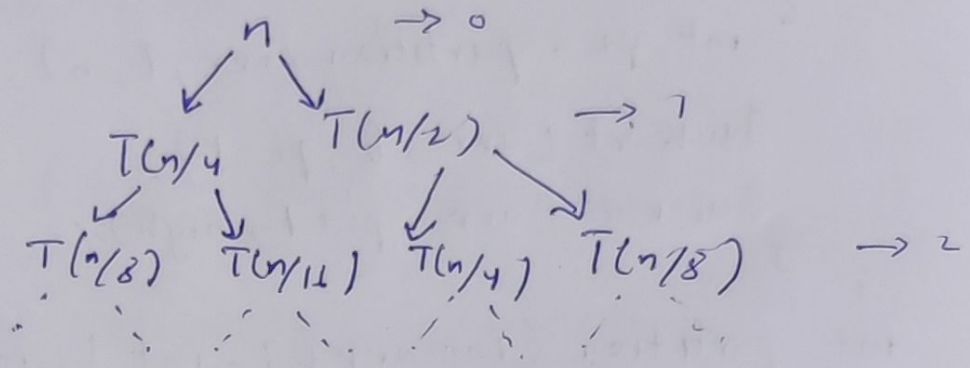
```
for (i=0; i<r1; i++)
{ for (j=0; j<c2; j++)
  { for (k=0; k<c1; k++)
    {
      rec[i][j] += a[i][k] * b[k][j];
    }
  }
}
```

3) log(log n)

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

Q.4 Solve the following recurrance relation

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



At level

$0 \rightarrow cn^2$

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

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

$\vdots$

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

$k = \log_2 n$

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

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

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

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

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

$\Rightarrow O(cn^2)$ Ans.

Q.5. Time complexity ?

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

→ for

i	j
1	1 + 3 + 5
2	1 + 4 + 7
3	1 + 5 + 9

$j = n-1$ // times

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

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

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

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

$$T(n) = O(n \log n) \quad \text{Ans.}$$

Q.6.

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

where k is a constant

Time complexity ?

for

i
2
2^k
$2k^2$
2^{k^3}
⋮
2^{k^m}

where

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

$k^m = \log_2 n$

$m = \log_k \log_2 n$

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

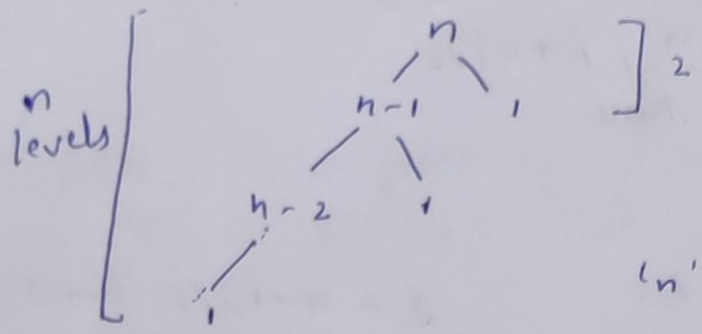
$1 + 1 + 1 \dots$ m times

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

Q.7.

Given algorithm divides away in 99% and 1% →

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



'n' work is done at each level

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

$= n \times n$

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

Lowest height = 2

highest height = n

$\therefore$ diff = $n - 2$ $\quad n > 1$

The given algorithm produces linear result.

Q.8 Arrange in increasing order of rate of growth:

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

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

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