Name :- Abhay Rawat

Course :- B.Tech (CSE)   Sec :- B

Sem :- V   Roll No :- 01

University Roll No :- 1961002

Design & Analysis of Algorithms

Tutorial Sheet 2

Ques 1 What is the Time Complexity of below code & how?

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

Ans $T(n) = O(k)$

$i = 0, 1, 3, 6, 10, \ldots\ldots, n$

$\cancel{S_n} = 1 + 3 + 6 + 10 + \ldots\ldots + n$

$\cancel{S_n} = \quad 1 + 3 + 6 + \ldots\ldots + (n-1) + n$

$(-) \quad (-) \quad (-) \quad (-) \quad (-) \quad (-)$

$0 = 1 + 2 + 3 + 4 + \ldots\ldots - n$

$n = 1+2+3+4+\ldots + k$ steps

$n = \frac{k}{2}[2 + (k-1)]$

$$2n = k^2 + k$$

$$2n + (1/2)^2 = (k + 1/2)^2$$

$$\sqrt{2n + (1/2)^2} = k + 1/2$$

$$k = \sqrt{2n + (1/2)^2} - 1/2$$

$$T(n) = O(k)$$

$$= O\left(\sqrt{2n + (1/2)^2} - 1/2\right)$$

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

Ques 2 Write recurrence relation for the recursive function that prints Fibnacci series. Solve the recurrence relation to get time complexity of the program. What will be the space complexity of this program & why?

Ans Recurrence Relation of Fibnacci series is

$$T(0) = 1$$

$$T(1) = 1$$

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

$$T(n-2) \approx T(n-1)$$

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

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

$$T(n) = 2\left[2T(n-2) + 1\right] + 1$$

$$= 2^2 T(n-2) + 2 + 1$$

$$T(n-2) = 2T(n-3) + 1$$

$T(n) = 2^2 [2T(n-3) + 1] + 2 + 1$

$T(n) = 2^3 T(n-3) + 2^2 + 2^1 + 2^0$

:

: General Term

$T(n) = 2^k T(n-k) + (2^0 + 2^1 + 2^2 + \cdots + 2^{k-1})$

~~T(n)~~ $T(n-k) = T(0)$

$n - k = 0$

$\boxed{n = k}$

$T(n) = 2^n T(0) + 2^0 + 2^1 + 2^2 + \cdots + 2^{n-1}$

$T(n) = 2^n (1) + \frac{2^0 (2^n - 1)}{2 - 1}$

$T(n) = 2^n + 2^n - 1$

$T(n) = 2^{n+1} - 1$

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

Space Complexity

If we draw the recursion tree of the fibonacii series then we found the max. height of tree will be n & hence the space complexity of fibonacii recursion will be O(n).

Ques 3 Write program which have complexity

(i) n log n.

```
int fun(int n)
{
  for(int i=0; i<n; i++)
  {
    for (int j=0; j<n; j=j*2)
      // O(1) operation
  }
}
```

(ii) n^3

Ans
```
int fun(int n)
{
  for(int i=0; i<n; i++)
  { for (int j=0; j<n; j++)
    { for (int k=0; k<n; k++)
        // O(1) operation
    }
  }
}
```

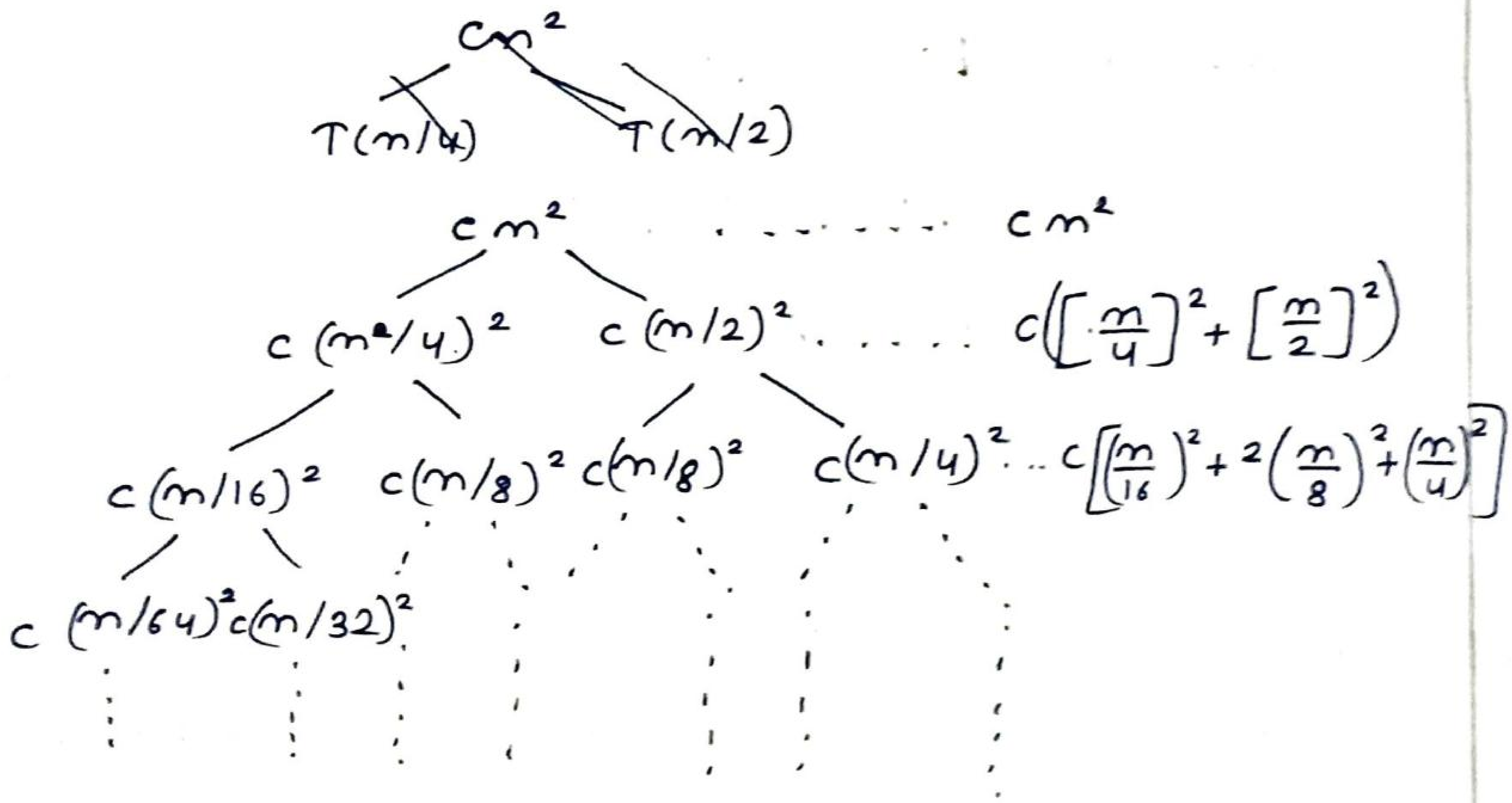
(iii) log(log n)

Ans
```
int funt(int n).
{
  for(int i≥0; i*i<n; i++)
    // O.(1) operation
}
```

Ques 4 Solve the following recurrence relation

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

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



$$T(n) = c\left[n^2 + \frac{5n^2}{16} + \frac{5^2 n^2}{256} + \frac{5^3 n^2}{16^3} + \cdots\right]$$

$$T(n) = cn^2\left[1 + \frac{5}{16} + \frac{5^2}{16^2} + \frac{5^3}{16^3} + \cdots\right]$$

$$T(n) = cn^2\left[\frac{1}{1 - 5/16}\right] \Rightarrow cn^2\left[\frac{16}{11}\right]$$

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

Scanned by TapScanner

Ques5 What is the time complexity of following function fun()?

Ans
```
int fun(int n)
{
  for(int i=1; i<=n; i++)   — n times
  {
    for (int j=1; j<n; j+=i) {
      // Some O(1) task
    }
  }
}
```

Ans

$i = 1$

$j = 1, 2, 3, 4 \ldots n$

n times

$i = 2$

$j = 1, 3, 5, 7 \ldots n$

$\frac{n}{2}$ times

$i = 3$

$j = 1, 4, 7, 10, \ldots n$

$\frac{n}{3}$ times

i	1	2	3	4	5	6
j	n	$n/2$	$n/3$	$n/4$	$n/5$	$n/6$

$$T(n) = n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \cdots + \frac{n}{n}$$

$$T(n) = n\left[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}\right]$$

$$T(n) = n \int_1^n 1/x \, dx$$

$$T(n) = n[\log x]_1^n \Rightarrow n \log n$$

Scanned by TapScanner

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

Ques 6 What should be the time complexity of

for(int i=2; i<=n; i=pow(i,k))

{

//some O(1) expression or statements

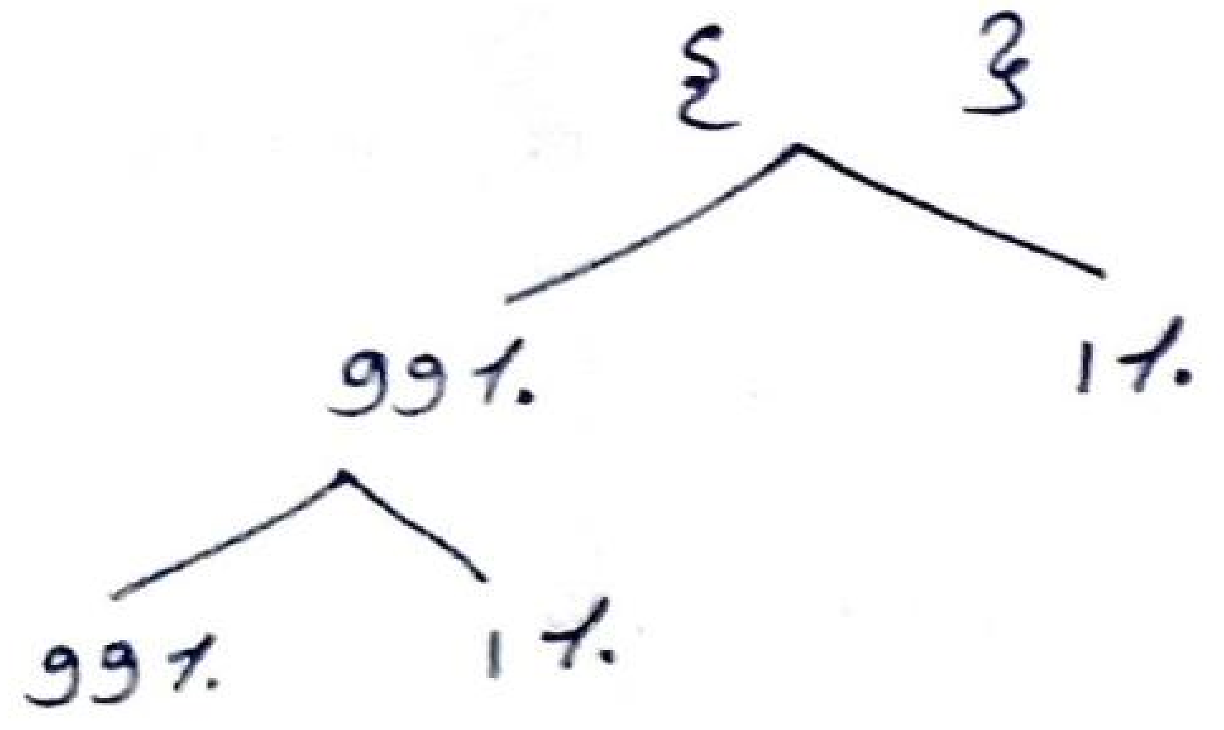
}

where k is an constant

Ans $O(\log(\log n))$

Ques 7 Write a recurrence relation ~~were~~ when quick sort repeatedly divide the array in to two parts of 99% & 1%. ~~Derve~~ Derive the time complexity in this case. Show the recursion tree while deriving time complexity & find the difference in heights of both the extreame parts. What do you understand by this analysis?

Ans $T(n) = T\left(\frac{99n}{100}\right) + T\left(\frac{n}{100}\right) + O(1)$



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

$$
\begin{array}{cccc}
 & & n & & + c \\
 & n-1 & & n-2 & + 2c \\
n-2 & n-3 & n-3 & n-4 & + 4c
\end{array}
$$

for each levels $2^n c$

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

Space complexity $= O(1)$

becz no new variable were created.

If we consider stack, recursively many function were ~~consider~~ called which are propotional to the depth of tree

$\therefore$ Space complexity $= O(n)$

Ques8 Arrange the following in increasing order of rate of groth.

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

Ans ~~100, log(log n), log n, n, n log n,~~

100, $\log(\log n)$, $\log n$, $\log 2n$, n, $n \log n$, n^2, 2^n, 2^{2^n}, 4^n, $\log n!$, $n!$

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

Ans 1, $\log(\log n)$, $\sqrt{\log n}$, $\log n$, $\log 2n$, n,

$2n$, $4n$, $n \log n$, n^2, 2^{2n}, & $\log(n!)$, $n!$

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

Ans 96, $\log_8 n$, $\log_2 n$, $5n$, $n \log_6 n$, $n \log_2 n$, $8n^2$, $7n^3$, 8^{2n}, $\log n!$, $n!$