11 – 18

Tutorial - 2

(11)

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

j	i
1	1
2	3
3	6
4	10

At i^{th} iteration, i is sum of integers till j.

$$\frac{K(K+1)}{2} >= n$$

$$K^2 >= n$$

$$K >= \sqrt{n}$$

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

12 Recurrence Relation

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

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

$= 2 * T(n-1)$

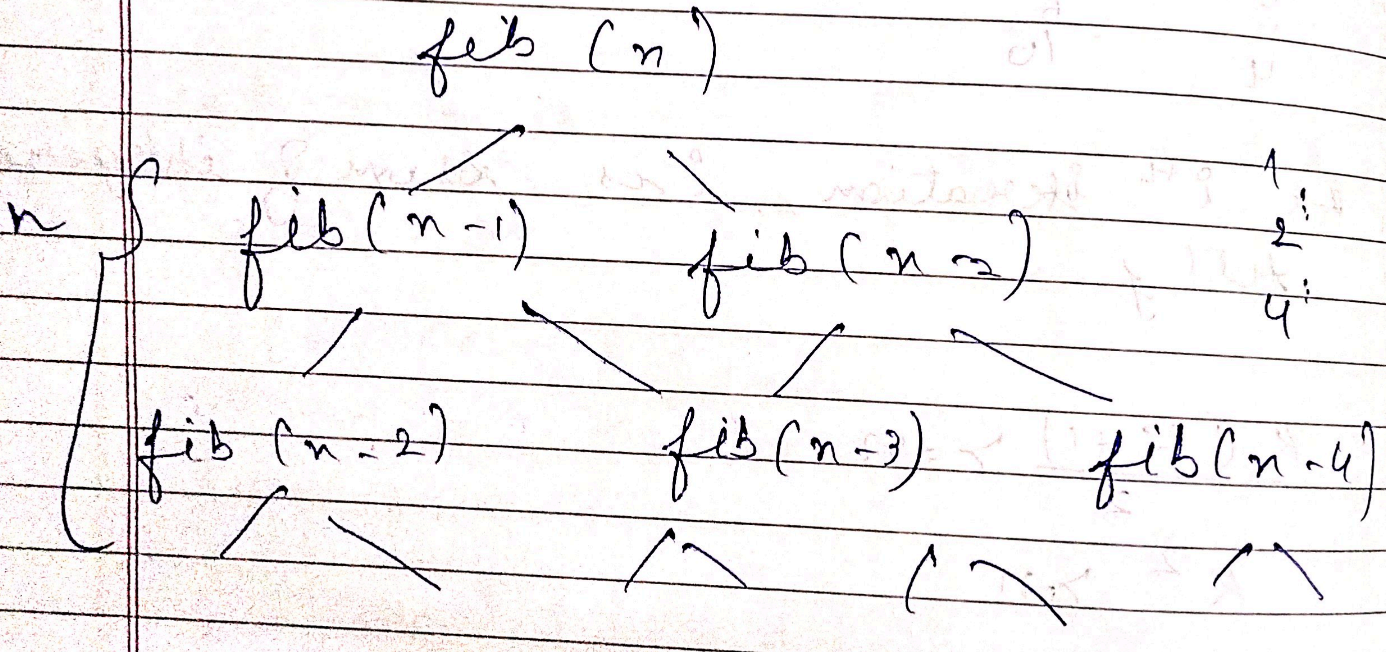
$= 2^2 * T(n-2)$

$\vdots$

$= 2^n * T(n-n)$

$\Rightarrow O(2n)$

$\Rightarrow O(1.68^n)$



S. C = O(n)

(13) n (log n) , n^3, log (log (n))

(1) n(log n)

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

(2) n^3

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

(13) $O(\log(\log n))$

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

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

$cn^2 \Rightarrow T(n)$

$T(n/4)$ $\quad$ $(T(n/2))$

$T(n/16)$ $\quad$ $T(n/8)$ $\quad$ $T(n/8)$ $\quad$ $T(n/4)$

$\cancel{T}(n) =)$ $\quad$ cn^2

$c\left(\frac{n}{4}\right)^2$ $\quad$ $c\left(\frac{n}{2}\right)^2$

$(n/16)^2$ $\quad$ $(n/8)^2$ $\quad$ $(n/8)^2$ $\quad$ $(n/4)^2$

Sum of 3 levels :

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

$$\Rightarrow c\left(n^2 + 5\left(\frac{n^2}{16}\right) + 25\, n^2/256 + \ldots\ldots n\right)$$

$\Rightarrow$ G.P : $r = 5/16$, $a = cn^2$.

$$S_n = \frac{a}{1 \pm r}$$

$$= \frac{cn^2}{1 - \frac{5}{16}} \Rightarrow \frac{c16n^2}{11}$$

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

(15)
```
int fun (int n)
{ for (int i=1; i<=n; i++)
  { for (int j=1; j<n; j+i)
    {}// O(1);
}}
```

$i = 1 \rightarrow n$ times

i^{th} iteration $\rightarrow n/i$ times

$$T(n) = O(n(1 + 1/2 + 1/3 + \dots))$$

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

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

{ //O(1);

}

} $k \rightarrow$ Constant

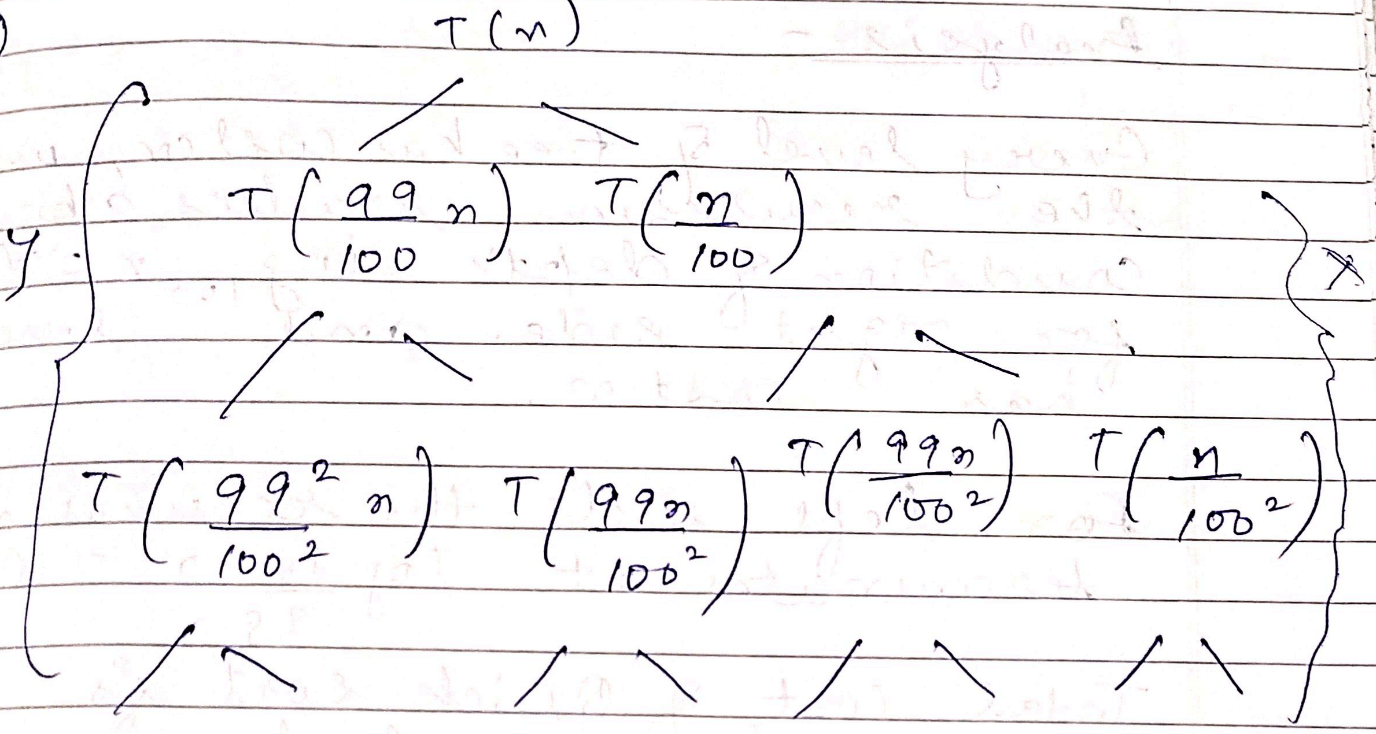
Sol: i takes $2, 2^k, (2^k)^k, \dots$

$2^{k^{\log_k \log(k)}}$

$\Rightarrow \sum \log_2^n = n \rightarrow$ last term

$T.C = O(\log \log n)$

17



$Y = \log_{\frac{100}{99}} n$ → height of left side.

$x = \log_{100} n$ → height of right side

Difference :

$$\log_{100} n - \log_{\frac{100}{99}} n$$

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

↳ Recurrence relation

Page

Analysis:-

Every level of tree has cost cn, until the recursion reaches a boundary condition of depth $\log_{100} n \Rightarrow O(\log n)$ for right side. and levels has cost n.

For left side the recursion terminates at $\log_{\frac{100}{95}} n = O(\log n)$

Total cost of Quick sort is

$\therefore$ $n \log n \Rightarrow O(n \log n)$.

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

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

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

Ashif