$\therefore$ $n_0 = 1$ & $c = 2$

Adesh Sengar
I, 14

Tutorials. 2

Q. 1 what is the T.C of below code and how.

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

Sln:- $i = 0, 1, 3, 6, 10, 15, 21, \ldots n$

k-terms

Let the sum of above k terms is S_k

$S_k = 1+3+6+10+15+21, \ldots + T_k$ — ①

$S_{k-1} = 1+3+6+10+15+21+\ldots + T_{k-1}$ — ②

Subtracting ② from ①

$T_k = S_k - S_{k-1} = 1+2+3+4+5+6+\ldots+k$

we have $T_k = n$

$\therefore 1+2+3+4+5+\ldots+k = n$

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

$k = \frac{-1 \pm \sqrt{8n+1}}{2}$

Taking only positive value we get total no. of times the loops runs for $i \geq k+1$

$= \sqrt{\frac{8n+1}{2}}$

$\therefore TC, T(n) = O\left(\left[\sqrt{\frac{8n+1}{2}}\right]^2\right) = O(n)$

Q.2 Write Recurrence Relation for the recursive function that prints fibonacci series. Solve the recurrence relation to get time complexity of the program. What will be the space complexity and why.

Recursive code:-

```
Int fib (int n)
{  if (n<=1) ——O(1)
      return n;
   return fib(n-1)+fib(n-2); →
                    T(n-1) + T(n-2)
```

Recurrance Relation:

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

On removing lowest order term $T(n-2)$ we get

$T(n) = T(n-1) + 1$ —— ①

Put $n = n-1$ in eq ①

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

put value of $T(n-1)$ in eq ①

$T(n) = T(n-2) + 1 + 1$ —— ②

put $n = n-2$ in eq ②

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

put the value of $T(n-2)$ in eq ②

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

on general,

$T(n) = T(k-3) + k$ — (3)

Q.3 Write programs which have complexity.

Sol:- 1. $n(\log n)$

```
for (i=1; i<=n; i++)
  { for (j=1; j<=n; j=j*2)
      { sum = sum+j;
      }
  }
```

2. n^3

```
for (i=0; i<n; i++)
  { for (j=0; j<n; j++)
    { for (k=0; k<n; k++)
        { sum = sum+k;
        }
    }
  }
```

3. $\log n(\log n)$

```
for (i=1; i<=n; i=i*2)
  { for (k=1; k<=n; k=k*2)
      { sum = sum+j;
      }
  }
```

Q.4 solve the recurrence Relation

$T(n) = T(\frac{n}{4}) + T(\frac{n}{2}) + cn^2$

Sol:- $T(n) = T(\frac{n}{4}) + T(\frac{n}{2}) + cn^2$

$\therefore T(\frac{n}{4}) \simeq T(\frac{n}{2})$

$\Rightarrow T(n) = 2T(\frac{n}{2}) + cn^2$

$a \geq 1$ & $b > 1$

$\therefore$ using master's method

$T(n) = aT(\frac{n}{b}) + f(n)$

$c = \log_b^a$

$c = \log_2^2 = 1$ $\quad f(n) > n^c$

$\therefore T(n) = O(f(n))$

$= O(n^2)$

5. What is the t.c of the following function.

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

Soln:-

for $i=1$; j is 1, 2, 3, 4, --- n-times

for $i=2$, j is 1, 3, 5, ---- $n/2$-times

for $i=3$, j is 1, 4, 7, ---- $n/3$-times

$T(n) = n + n/2 + n/3 + \cdots n/4 + \cdots$

$n(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots)$

$n \int_1^n dx/x$

$= (\log x)_1^n$

T.C $= n \log n$

Q.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 a constant.

Soln:- for first iteration $i = 2$

second iteration $i = 2^k$

third " $i = (2^k)^k = 2^{k^2}$

⋮

n^{th} iteration, $i = 2^{k^c}$ loop ends at $2^{k^c} = n$

apply $\log n = \log_2 k^c$

$k^c = \log n$

$c = \log_e(\log n)$

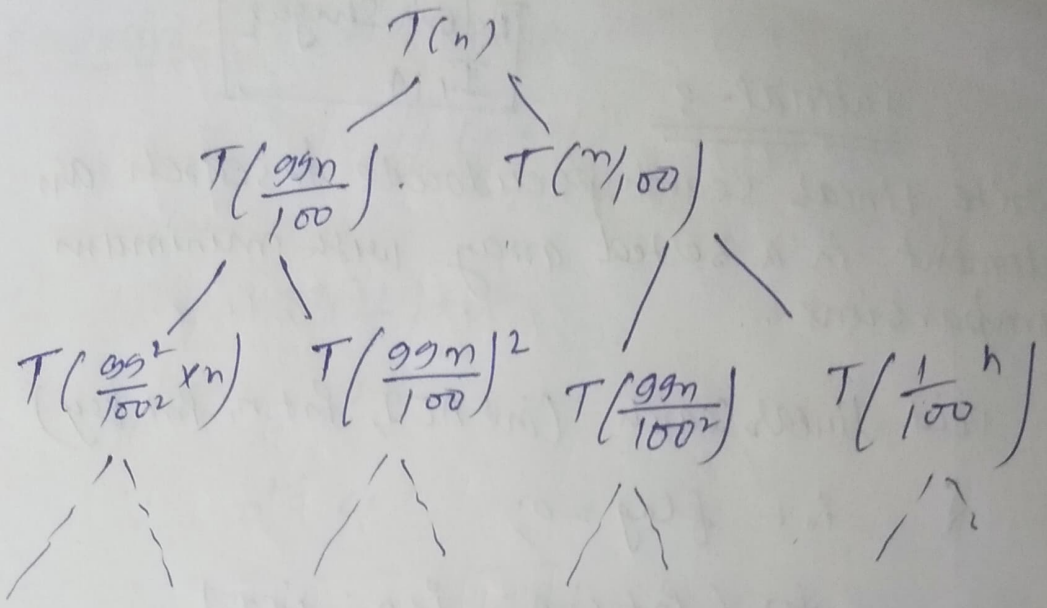
(7) Write a recurrence relation when quicksort repeatedly divides the array in to two parts of 99% and 1%. Derive the T.C in this case. Show the recursion tree while deriving T.C and find the difference in heights of both the extreme parts. what do you understand by this analysis.

Soln:- 99 to 1 in quicksort

when pivot is where from front or end always.

So, $T(n) = T(99n/100) + T(n/100) + O(n)$

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



n

$$n = \left(\frac{95}{100}\right)^k$$

$$\log n = k \log\left(\frac{95}{100}\right)$$

$$k = \log_n\left(\frac{100}{99}\right)$$

$$T.C = n^x \log_{\frac{100}{95}}(n)$$

(8). Arrange the following in increasing order of rate of growth.

Sol:- a. $1 < \log\log(n) < \log^2 n < \log n < \log n! < n <$

$n \log n < n^2 < 2^n < 4^n < 2^n < 2^{n^n} < n!$

b. $1 < \log(\log n) < \sqrt{\log n} < \log(n) < 2\log(n) <$

$\log(2n) < n < 2n < 4n < \log n! < n \log(n) <$

$2(2^n)$