Tutorial - 2

Ans-1 -

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

	0	1	2	3 ...		(n-1)
i	1	3	5	7	...	n(n-1)
j	1	2	3	4	...	n

$i = 1, 3, 5, 7 \ldots\ldots n(n-1)$

$T(C)$ of j $(1, 2, 3, 4 \ldots n) = O(n)$

sum of A.P. i.e. $(i) = \frac{n(n+1)}{2}$

$\Rightarrow \frac{n(n+1)}{2} + n$ (for j) $< n$

$\Rightarrow \frac{n(n+1)}{2} + n$ (for j) $< n$

$\Rightarrow \underline{n^2 + n + 2n} < 2n$

$\Rightarrow n^2 < 2n \Rightarrow n^2 < n$

$\Rightarrow n < \sqrt{2}$

Chitra

$T(c) = O(\sqrt{n})$ Ans

Ans 2 - Fib series program using Recursion :-

```
# include <bits/stdc++.h>
using namespace std.;
int fib (int n)
{
   if (n<=1)
return n;
  return fib (n-2) + fib (n-2) ;
}
   int main (1)
   {
     int n=0 ;
   count << fib (n);
     getchar (1);
      return 0;
   }
```

$T(c) = T(n)$ i.e. linear

Ans 3 - (i) n (log n) :-

```
for (i=0 ; i<n ; i++)
{
 for (j=1 ; j<n ; j=j*2)     n x log n
}
                    -> n(log n)
}
```

Date ___/___/___

ii) n^3 :- Floyd Warshall Algorithm
shortest path b/w vertices

i.e, for k = 1 to n
for i = 1 to n
" j = 1 to n "
$A^k(i,j) = \min(A^{k-1}[i,j], A^{k-1}[i,k] + A^{k-1}(k,j])$

return A

iii) for (i = 1 ; i < n ; i = i * 2) (T(c) = log (log A))
{
p++ ;
}
for (j = 1 ; j < p ; j = j * 2)
{
}

Ans 4 - T(c) of $T(n) = T(n/4) + T(n/2) + (n)^2$

assuming $T(n/2) >= T(n/4)$
On Rewriting
$T(n) <= 2T(n/2) + (n)^2$

Now applying Master's theorem to RHS
$T(n) <= \Theta(n)^2 \Rightarrow T(n) = O(n^2)$

$T(n) >= n^2 \Rightarrow Tn >= \Theta(n)^2 \Rightarrow T(\Omega) = \Omega$

$\therefore T(n) = \Theta(n^2)$ & $T(n) = \Omega(n^2)$

$\Rightarrow T(n) = \Theta(n^2)$ Ans

Q5 - What's the complexity of following fun()?

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

Ans - for $i=1$, inner loop executes n times
$i=2$, inner " " $n/2$ times
$i=3$, " " $n/3$ times
$i=n$; " " n/n times

$T(c)$ of $(n + \frac{n}{2} + \frac{n}{3} + \dots n/n)$

$\Rightarrow n^* \left(1/1 + \frac{1}{2} + \frac{1}{3} + \dots \frac{1}{n}\right)$

$= O(\log n)$ Ans

Chitra

Date ___/___/______ Page No.: ________

Q6 - What should be time complexity of

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

where k = constant

Ans - i takes values $2, 2^k, (2^k)^k = 2^{k^2}$

$(2^{k^2})^k = 2^{k^3} \ldots 2^{k^{\log_k(\log(n))}}$

Last term must <= n

$\rightarrow 2^{k^{\log_k[\log(n)]}} = 2^{\log(n)} = n$

equiv to last term

$\therefore$ there are $\log_k(\log(n))$

$\therefore T(C) = O(\log(\log(n)))$ Ans

Q7 - Write recursive relation . . . what do you understand by analysis?

Ans 7 - $T(n) = T\left(\frac{9n}{10}\right) + T\left(\frac{n}{10}\right) + O(n)$

Taking one branch 99% & other 1%

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

1^{st} level = n

Chitra

2^{nd} level $= \frac{99}{100} + \frac{n}{100} = n$

So, 3^{nd} level remains same for any kind of position

Larger branch $= O(n \log 100/99n)$

For shorter branch

$\hookrightarrow = \Omega(A \log_{10} n)$

either base complexity of $O(n(\log n))$ remains

Ans8 -

a) $100 < \sqrt{n} < \log(\log n) < \log(n) < n < n \log n$

$< \log n^{1} < n^2 < n-1 < 2^n < 4^n < 2^{2n}$

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

$n \log = < 2n < 4n = 2(2^n) < n ; 2n^2$

c) $96 < \log_e n < \log n! < n \log_2 n < n \log_e n < 5n$

$< n! < 8n^2 < 7n^3 < 8^{n^{2n}}$