Experiment -03

i) Write a 'C' program to calculate time & space complexity

ii) Calculate time complexity when there are many if-else statements inside loops

→

for example we take function fun() with 2 for loops

```
fun()
{ for (i=0; i<n; i++) { for j=0; i<n; j++ {
    // if else statement // } }
```

loop1 = O(n)
loop2 = O(n)
if else st. = O(1)
⇒ O(n × n × 1)
= n^2

ii) what is the time complexity of fun()?

```
int fun(int n)
  int count = 0;
for (int i=0; i<n; i++)   //n          } n^2
  { for(int j = i = i; j>0; j--) //n
      {
          count = count + 1;   // O(1)
        return (count; } }
```

loop1 → O(n)
loop2 → O(n)
loop3 → O(1)
(n × n) × O(1)
n^2

∴ Time complexity of thinct fun()
$f(n) = n^2$

iii) What is time complexity of following function fun()

```
void fun (int n)
  { int i, j;
   for (i=1; i<=n; i++)        //n
     { for(j=1 ; j<=log(i) ; j++) // log(i)
         { printf ("Welcome"); // O(1)
     } }
```

loop 1 → O(n)
loop 2 → O(log i)
loop 3 → O(1)
O (n x log i x 1)
O (n log i)
n log i

The time complexity for this function is :- ~~for~~
$f(n) = n \log i$

As the outer loop runs from 1 to N
inner loop runs from 1 to log (i)
log (i) = log (1) + log (2) + ... + log (N)
this can be written as log (N)

So the time complexity can be written as
(log n)n

8m
29/8/23