Page No.
Date

Assignment-02

Q1

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

n = 5

i	j
0	1
0+1	2
0+1+2	3
~~0+1+2+3~~	

$\frac{n(n+1)}{2}$

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

Q2

```
~~void~~ int fiboracci (int n)
  a=0, b=1
  if (n<=1 ~~|| n==2~~)
    return 1;
  fiboracci (n-1) + fib (n-2);
```

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

$T(n) = 1 + 2 + 4 + 8 \dots 2^n$

$= \frac{2^{n+1}-1}{2-1} = 2^n \cdot 2 - 1$

$O(2^n)$

Page No.
Date

The complexity of this fun is O(n).
We will have n stack frame, f(n), f(n-1), f(n-2) ... f(1),
and O(1) space per stack frame to store arg. (n).

i) O(n log n) -
```
void fun (int n)
{ int c = 0
  for (i = 0; i < n; i++)
  { for (j = 0; j < n; j += 2)
    { c++; }
  }
}
```

ii) O(n³) -
```
void fun (int n)
{ int c = 0;
  for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
      for (k = 0; k < n; k++)
        c++;
}
```

iii) O(log(log n)) :-
```
for (i = 2; i <= n; i = pow(i, c))
{
  // some O(1) expression
}
```

Page No.
Date

Q4 $T(n) = T(n/4) + T(n/2) + cn^2$

Q5 $O(n^3)$

Q6 $O(\log(\log n))$

Q7