Name: Arijan Hossain
ID: 20101099
Sec: 09
Course: CSE 221 Lab

Lab 01

Answer 2

For Implementation-1,

```
def fibonacci_1(n):                          ⟶ T(n)
    if n<=0:                                 ⟶ O(1)
        print("Invalid input!")
    elif n<=2:                               ⟶ O(1)
        return n-1
    else:
        return fibonacci_1(n-1)+fibonacci_1(n-2)  ⟶ T(n-1) + T(n-2)
```

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

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

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

for large value of n,

$T(n-1) \approx T(n-2)$

$T(n) = 2T(n-1) + 1$ —(1)

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

$= 2^2 T(n-2) + 2 + 1$ —(2)

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

$= 2^3 T(n-3) + 2^2 + 2 + 1$ —(3)

$= 2^{n+1} T(1) + 2^n + 2^{n-1} \ldots$

$= 2^{n+1} + 2^n + 2^{n-1} \ldots$

$= 2^0 + 2^1 + 2^2 \ldots$

$T(n) = 2^{n+1} - 1 = O(2^n)$

∴ Time Complexity = $O(2^n)$ (Ans:)

For Implementation-2,

```
def fibonacci_2(n):
    fibonacci_array = [0, 1]  ⟶  1
    if n<0:  ⟶ 1
        print("Invalid input!")  ⟶  1
    elif n<=2:  ⟶ 1
        return fibonacci_array[n-1]  ⟶  1
    else:
        for i in range(2,n):  ⟶ n
            fibonacci_array.append(fibonacci_array[i-1]+
                                   fibonacci_array[i-2])   } 1 x n = n
    return fibonacci_array[-1]  ⟶ 1
```

Considering worst case complexity, the code will go through the else condition.

$$f(n) = 1 + 1 + 1 + n + n + 1$$

$$= 2n + 4$$

$$= O(n)$$

∴ Time complexity = $O(n)$ (Ans:)

Answer 4

Time Complexity for Multiplication part:

$C = [[0]*(n) \text{ for } i \text{ in range}(n)] \longrightarrow n$

for i in range(n): $\longrightarrow n$

  for j in range(n): $\longrightarrow n \times n = n^2$

    for k in range(n): $\longrightarrow n \times n \times n = n^3$

      $C[i][j] += A[i][k] * B[k][j] \longrightarrow n \times n \times n \times 1 = n^3$

$f(n) = n + n + n^2 + n^3 + n^3$

$f(n) = 2n^3 + n^2 + 2n$

$= O(n^3)$

∴ Time complexity = $O(n^3)$ (Ans:)