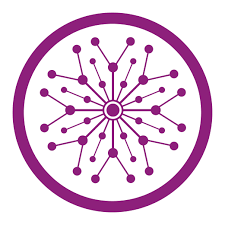
**

**The Superior University**

|  |  |  |
| --- | --- | --- |
| Name:Mohid Naghman | Roll No:049 | Course: DSA Lab |
| Semester:3 | Section:3A | Department: |
| Submitted To: | Total Marks: 10 | Date: |

**Lab Task 05**

**Time Complexity**

**Find the time complexity of the below programs:**

**Code:**

**Bubble Sort Program**

def bubble\_sort(arr):

    n = len(arr)

    for i in range(n):

        for j in range(0, n-i-1):

            if arr[j] > arr[j+1]:

                arr[j], arr[j+1] = arr[j+1], arr[j]

*# Test the function*

arr = [64, 34, 25, 12, 22, 11, 90]

bubble\_sort(arr)

print("Sorted array is:", arr)



**Factorial Program:**

def factorial(n):

    result = 1

    for i in range(1, n + 1):

        result \*= i

    return result

*# Test the function*

number = 5

print("Factorial of", number, "is", factorial(number))

****

**Fibonacci Sequence Recursive Program:**

def fibonacci(n):

    if n <= 1:

        return n

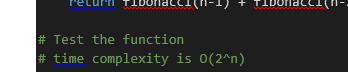
    return fibonacci(n-1) + fibonacci(n-2)

*# Test the function*

num\_terms = 10

for i in range(num\_terms):

    print(fibonacci(i), end=" ")

****

**Merge Sort Program**

def merge\_sort(arr):

    if len(arr) > 1:

        mid = len(arr) // 2

        left\_half = arr[:mid]

        right\_half = arr[mid:]

        merge\_sort(left\_half)

        merge\_sort(right\_half)

        i = j = k = 0

        while i < len(left\_half) and j < len(right\_half):

            if left\_half[i] < right\_half[j]:

                arr[k] = left\_half[i]

                i += 1

            else:

                arr[k] = right\_half[j]

                j += 1

            k += 1

        while i < len(left\_half):

            arr[k] = left\_half[i]

            i += 1

            k += 1

        while j < len(right\_half):

            arr[k] = right\_half[j]

            j += 1

            k += 1

*# Test the function*

arr = [38, 27, 43, 3, 9, 82, 10]

merge\_sort(arr)

print("Sorted array is:", arr)



**Prime Numbers Program:**

def is\_prime(n):

    if n <= 1:

        return False

    for i in range(2, int(n \*\* 0.5) + 1):

        if n % i == 0:

            return False

    return True

*# Test the function*

number = 29

print(number, "is prime?" , is\_prime(number))

****