# Lab 2: Asymptotic Complexity of an Algorithm

## Task 1:

Answer the following questions regarding the time complexity of various expressions and visualize their growth.

### a. Is (n+5)^2 = O(n log n)?

No, (n+5)^2 expands to n^2 + 10n + 25, which is primarily dominated by n^2. Quadratic growth (n^2) is much faster than n log n, making (n+5)^2 not fit within O(n log n).

### b. Is n^(14/16) = O(n log n)?

No, n^(14/16) simplifies to n^0.875. Since any polynomial growth n^k where k > 0 grows faster than n log n, n^0.875, being a fractional polynomial, does not fit within O(n log n).

### c. What is the time complexity of n+5, (n+5)^3, and (n+5)^5?

The time complexity of n+5 is O(n), reflecting linear growth. For (n+5)^3 and (n+5)^5, the complexities are O(n^3) and O(n^5), respectively, indicating polynomial growth rates.

### d. Code for Visualizing Growth Rates

Python code is provided to plot the growth rates of (n+5)^2, n^(14/16), n+5, (n+5)^3, and (n+5)^5 on the same graph, demonstrating their behavior as n increases from 1 to 50.

import matplotlib.pyplot as plt  
import numpy as np  
n = np.arange(1, 51)  
plt.figure(figsize=(10, 6))  
plt.plot(n, (n+5)\*\*2, label='(n+5)^2')  
plt.plot(n, n\*\*(14/16), label='n^(14/16)')  
plt.plot(n, n+5, label='n+5')  
plt.plot(n, (n+5)\*\*3, label='(n+5)^3')  
plt.plot(n, (n+5)^5, label='(n+5)^5')  
plt.yscale('log')  
plt.xlabel('n')  
plt.ylabel('Function value')  
plt.title('Growth Rates Comparison')  
plt.legend()  
plt.grid(True)  
plt.show()

## Task 2:

Analyze the given pseudocode to determine the number of times 'Ping' is executed and compute the time complexities.

### 1. Nested Loop Pseudocode

for j in 0...n do  
 for k in 0...j+1 do  
 Ping

Analysis: This nested loop structure results in Ping being executed approximately (n\*(n+3)/2) times, leading to O(n^2) complexity.

### 2. While Loop Pseudocode

i = 1  
while i <= n do  
 Ping  
 j = n  
 while j > i do  
 Ping  
 j -= 1  
 i += 1

Analysis: The outer loop runs n times, and the inner loop runs decreasingly fewer times as i increases, resulting in a total complexity of O(n^2).

### 3. Increasing Inner Loop Pseudocode

Ping  
i = 1  
while i <= n do  
 Ping  
 j = 1  
 while j <= i\*i do  
 Ping  
 j += 1  
 i += 1

Analysis: The inner loop's limit grows as the square of i, leading to a complexity of O(n^3) due to the summation of squares up to n.