МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ

ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

НАВЧАЛЬНА ДИСЦИПЛІНА

«АЛГОРИТМИ І СТРУКТУРИ ДАНИХ»

ЗВІТ

З ПРАКТИЧНОЇ РОБОТИ №2

Виконала:

студентка групи КН-24-1

Процко П.Д.

Перевірив:

доцент кафедри АІС

Сидоренко В. М.

Кременчук 2025

## **Тема**. Асимптотична складність алгоритмів. Інші нотації

## **Мета**: набути практичних навичок у розв’язанні задач на оцінку асимптотичної складності алгоритмів у Ω, Θ, 𝜊, 𝜃, 𝜔-нотаціях

Хід роботи

**Задачі для самостійного розв’язання**

Виконати індивідуальне завдання. Завдання полягає у розв’язанні двох задач, які потрібно вибрати зі списку, наведеного нижче. Правило вибору номерів наступний: 𝑛, 𝑛 + 5, де 𝑛 – номер студента в списку групи. У разі, якщо було досягнуто кінця списку задач, потрібно циклічно повернутися на його початок

### **Завдання 1**

**Умова:** Маємо дві функції f(n) = 4n³ - 12n + 9, g(n) = n³. Довести, що f(n) = Ω(g(n)).

**Розв'язання:** f(n) = 4n³ - 12n + 9 = n³ + (3n³ - 12n + 9)

n = 2: 3·2³ - 12·2 + 9 = 3·8 - 24 + 9 = 24 - 24 + 9 = 9 > 0

Отже, для n₀ = 2: f(n) = 4n³ - 12n + 9 ≥ n³ = 1·g(n)

Таким чином, f(n) = Ω(g(n)) з константами c = 1 і n₀ = 2.

### **Завдання 6**

**Умова:** Нехай f(n) = 6n³ - 9n² + 12n і g(n) = n³. Довести, що f(n) = Ω(g(n)).

**Розв'язання:** f(n) = 6n³ - 9n² + 12n = n³ + (5n³ - 9n² + 12n)

5n³ - 9n² + 12n = n²(5n - 9) + 12n

При n = 2: 5·2 - 9 = 10 - 9 = 1 > 0, отже n²(5n - 9) > 0 для n ≥ 2. Також 12n > 0 для n ≥ 1.

Таким чином, для n₀ = 2: f(n) = 6n³ - 9n² + 12n ≥ n³ = 1·g(n)

Отже, f(n) = Ω(g(n)) з константами c = 1 і n₀ = 2.

# 

# **Висновок**

У Практичній роботі досліджено асимптотичну складність алгоритмів через різні нотації. Розв'язано два завдання з доведенням приналежності функцій до класу Ω(g(n)). Використано метод перетворення функцій для знаходження констант c і n₀.

## **Відповіді на контрольні запитання:**

1. **Що таке асимптотична складність алгоритму?** Асимптотична складність алгоритму - це математична характеристика, що описує поведінку алгоритму при збільшенні розміру вхідних даних, зосереджуючись на темпі зростання функції складності (часової або просторової) при наближенні розміру вхідних даних до нескінченності, ігноруючи константи та члени нижчого порядку.
2. **Які інші нотації, крім O-нотації, використовуються для вираження асимптотичної складності?** Крім O-нотації використовуються: Ω (велика омега) - для нижньої межі, Θ (велика тета) - для точної оцінки, o (мала о) - для строгої верхньої межі, ω (мала омега) - для строгої нижньої межі, а також іноді використовуються нотації O~, Õ (м'яке О) і ≈.
3. **Як визначити асимптотичну складність алгоритму за допомогою символів Θ і Ω?** Θ-нотація: f(n) = Θ(g(n)), якщо існують додатні константи c₁, c₂ і n₀, такі що c₁·g(n) ≤ f(n) ≤ c₂·g(n) для всіх n ≥ n₀. Це означає, що функція f(n) обмежена зверху і знизу функцією g(n).  
     
    Ω-нотація: f(n) = Ω(g(n)), якщо існують додатні константи c і n₀, такі що f(n) ≥ c·g(n) для всіх n ≥ n₀. Це означає, що функція f(n) обмежена знизу функцією g(n).
4. **Яка різниця між O-нотацією, Θ-нотацією і Ω-нотацією?** O-нотація визначає асимптотичну верхню межу, тобто f(n) = O(g(n)) означає, що функція f зростає не швидше, ніж функція g.  
     
    Ω-нотація визначає асимптотичну нижню межу, тобто f(n) = Ω(g(n)) означає, що функція f зростає не повільніше, ніж функція g.  
     
    Θ-нотація визначає асимптотично точну оцінку, тобто f(n) = Θ(g(n)) означає, що функція f має той же порядок зростання, що й функція g. Фактично, f(n) = Θ(g(n)) тоді і тільки тоді, коли f(n) = O(g(n)) і f(n) = Ω(g(n)) одночасно.
5. **Які основні властивості інших нотацій, таких як o (маленька о), ω (маленька омега)?** o-нотація (мала о) визначає строгу верхню межу: f(n) = o(g(n)), якщо для будь-якої константи c > 0 існує n₀, таке що f(n) < c·g(n) для всіх n ≥ n₀. Це означає, що функція f зростає строго повільніше, ніж функція g, на відміну від O-нотації, де допускається рівність швидкостей зростання.  
     
    ω-нотація (мала омега) визначає строгу нижню межу: f(n) = ω(g(n)), якщо для будь-якої константи c > 0 існує n₀, таке що f(n) > c·g(n) для всіх n ≥ n₀. Це означає, що функція f зростає строго швидше, ніж функція g, на відміну від Ω-нотації, де допускається рівність швидкостей зростання.  
     
    Також існує зв'язок: f(n) = o(g(n)) тоді і тільки тоді, коли g(n) = ω(f(n)), і f(n) = O(g(n)) тоді і тільки тоді, коли g(n) = Ω(f(n)).