

Практична робота №1

Тема. Асимптотична складність алгоритмів. **O**-нотація

Мета: набути практичних навичок у розв'язанні задач на оцінку асимптотичної складності алгоритмів у O .

Завдання

8. Довести, що $f(n) = 2n^2 + 10n + 3 = O(n^2)$.

Доводимо:

Функція $f(n) = 2n^2 + 10n + 3$.

$f(n) \leq c * n^2$ для всіх $n \geq n_0$

$2n^2 + 10n + 3 \leq (2 + 10 + 3)n^2 = 15n^2$

$f(n) \leq 15n^2$ для всіх $n \geq n_0$

$f(n) = O(n^2)$.

13. Задано функції $f(n) = 100n^3 + 8$ та $g(n) = n^3$. Доведіть, що $f(n) = O(g(n))$.

$f(n) \leq c * g(n)$ для всіх $n \geq n_0$, де $g(n) = n^3$.

$f(n) = 100n^3 + 8 \leq (100 + 8)n^3 = 108n^3$

$f(n) \leq 108n^3$ для всіх $n \geq n_0$

$f(n) = O(n^3)$.

Контрольні питання

1. Що таке асимптотична складність алгоритму?

Асимптотична складність алгоритму — це математична характеристика, яка описує поведінку часу виконання або використання пам'яті алгоритмом при збільшенні розміру вхідних даних. Вона дозволяє оцінити ефективність алгоритму в контексті великих обсягів даних і допомагає порівнювати алгоритми, незалежно від конкретних апаратних засобів або реалізацій.

2. Яким чином визначається O -нотація і яка її сутність?

O -нотація (велике O) використовується для опису верхньої межі швидкості росту часу виконання або використання пам'яті алгоритмом. Її сутність полягає в тому, щоб вказати, як швидко збільшується складність алгоритму відносно розміру вхідних даних у найгіршому випадку. Наприклад, якщо алгоритм має

асимптотичну складність $O(n^2)$, це означає, що час виконання алгоритму буде збільшуватися не більше, ніж квадратично з ростом розміру вхідних даних.

3. Які основні правила використання O -нотації при аналізі алгоритмів?

Основні правила використання O -нотації при аналізі алгоритмів:

- Константи та менші члени ігноруються: Якщо є кілька складових з різними степенями, залишають тільки домінуючий член, а константи ігноруються. Наприклад, якщо алгоритм має складність $3n^2+5n+2$, його O -нотація буде $O(n^2)$.
- Асимптотична поведінка враховується для великих даних: Вона визначає продуктивність алгоритму, коли розмір вхідних даних стає дуже великим.
- Описує найгірший випадок: O -нотація, зазвичай, використовується для оцінки продуктивності алгоритму у найгіршому сценарії.

4. Що означають вирази $O(1)$, $O(n)$, $O(n^2)$ в контексті асимптотичної складності?

Значення виразів в контексті асимптотичної складності:

- $O(1)$: Константна складність — час виконання або використання пам'яті не залежить від розміру вхідних даних.
- $O(n)$: Лінійна складність — час виконання або використання пам'яті пропорційний розміру вхідних даних.
- $O(n^2)$: Квадратична складність — час виконання або використання пам'яті пропорційний квадрату розміру вхідних даних.

5. Яким чином визначити асимптотичну складність алгоритму за його кодом або математичним виразом?

Щоб визначити асимптотичну складність алгоритму за його кодом або математичним виразом:

- Проаналізуйте цикли в коді: Найчастіше складність залежить від вкладеності циклів. Наприклад, один цикл зазвичай має складність $O(n)$, а вкладений цикл — $O(n^2)$.
- Врахуйте рекурсії: Для рекурсивних алгоритмів часто використовуються рекурсивні рівняння, які допомагають визначити їх складність.
- Ігноруйте константи та менш значущі члени: Залиште лише домінуючий член, що найшвидше зростає зі збільшенням розміру вхідних даних.