Практична робота № 2

Тема. Асимптотична складність алгоритмів. Інші нотації Мета: набути практичних навичок у розв'язанні задач на оцінку асимптотичної складності алгоритмів у Ω, Θ, o, θ, ω-нотаціях.

Постановка завдання. 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.

№14

Щоб показати, що $f(n)=O(g(n))$f(n)=O(g(n)) для функцій $f(n)=n4-2n3+3n+7$f(n)=n4−2n3+3n+7 та $g(n)=n4$g(n)=n4, використаємо метод меж (також відомий як критерій Ліміта).

- Обчислимо границю:
- lim⁡$n\to\infty f(n)/g(n)=$lim⁡$n\to\infty(n^4-2n^3+3n+7)/(n4)$
- Спрощуємо вираз:
- lim⁡$n\to\infty(n^4-2n^3+3n+7)/n$^4=lim⁡$n\to\infty((n^4/n^4)-(2n^3/n^4)+(3n/n^4)+($ $7/n^4))$
- Розділимо на окремі дроби: lim⁡$n\to\infty(1-(2n^3/n^4)+(3n/n^4)+(7/n^4))=$lim⁡$n\to\infty(1-(2/n)+(3/n^3)+(7/n$ ^4))

Таким чином:

lim⁡$n\to\infty(1-(2/)n+(3/n^3+(7/n^4))=1-0+0+0=1$

Тепер знаходимо границю кожного доданка окремо:

- Границя *1*1 залишається 1, так як це константа.
- Границя *2n*n2 прямує до 0 при *n→∞*n→∞.
- Границя *3n3*n33 прямує до 0 при *n→∞*n→∞.
- Границя *7n4*n47 прямує до 0 при *n→∞*n→∞.

Оскільки границя є скінченним числом (1), ми можемо зробити висновок, що:

$f(n)=O(g(n))$

Таким чином, ми показали, що $f(n)=O(g(n))$f(n)=O(g(n)) за допомогою методу меж.

Завдання №4

Щоб довести, що $f(n)=\Omega(g(n))$f(n)=Ω(g(n)) для функцій $f(n)=2n3+7n2-4$f(n)=2n3+7n2−4 та $g(n)=n3$g(n)=n3, ми використовуємо визначення нотації ΩΩ. За визначенням, $f(n)=\Omega(g(n))$f(n)=Ω(g(n)) означає, що існують такі константи $c>0$c>0 і $n0$n0, що для всіх $n\geq n0$n≥n0 виконується:

$f(n)\geq c\cdot g(n)$f(n)≥c·g(n)

Формально, ми повинні знайти такі константи cc та $n0$n0, щоб нерівність була виконана. Для цього спростимо вирази функцій і знайдемо підходящі константи.

$f(n)=2n^3+7n^2-4$

$g(n)=n$^3

$2n^3+7n^2-4\geq c\cdot n$^3

$2n^3+7n^2-4\geq n$^3

$2n^3-n^3+7n^2-4\geq 0$

$n3+7n2-4\geq 0$

1^3+7·1^2−4=1+7−4=4≥0

Таким чином, ми довели, що $f(n)=\Omega(g(n))$:

$f(n)=2n^3+7n^2-4\geq 1\cdot n$^3

Це підтверджує, що $f(n)=\Omega(n/3)$.

Контрольні питання :

1. Що таке асимптотична складність алгоритму?

Асимптотична складність алгоритму – це міра ефективності алгоритму, яка показує, як змінюється час виконання або обсяг використаної пам'яті алгоритму в залежності від розміру вхідних даних, коли цей розмір прямує до нескінченності. Вона допомагає порівнювати алгоритми незалежно від апаратних засобів чи реалізаційних деталей.

2. Які інші нотації, крім O-нотації, використовуються для вираження асимптотичної складності?

Крім O-нотації (Big O), використовуються ще такі нотації:

- **Θ-нотація (Theta)**: Описує точну асимптотичну поведінку алгоритму, тобто як нижню, так і верхню межу.
- **Ω-нотація (Omega)**: Описує нижню межу асимптотичної складності.
- **o-нотація (маленька o)**: Описує верхню межу, яка не є точним верхнім обмеженням.
- **ω-нотація (маленька омега)**: Описує нижню межу, яка не є точним нижнім обмеженням.

3. Як визначити асимптотичну складність алгоритму за допомогою символів Θ і Ω?

Визначення складності за допомогою Θ-нотації:

Функція $f(n)f(n)$ належить до $\Theta(g(n))\Theta(g(n))$, якщо існують такі константи $c1>0c1>0$, $c2>0c2>0$ та $n0n0$, що для всіх $n\geq n0n\geq n0$ виконується:

$c1\cdot g(n)\leq f(n)\leq c2\cdot g(n)c1\cdot g(n)\leq f(n)\leq c2\cdot g(n)$

Визначення складності за допомогою Ω-нотації:

Функція $f(n)f(n)$ належить до $\Omega(g(n))\Omega(g(n))$, якщо існують константи $c>0c>0$ та $n0n0$, що для всіх $n\geq n0n\geq n0$ виконується:

$f(n)\geq c\cdot g(n)f(n)\geq c\cdot g(n)$

4. Яка різниця між O-нотацією, Θ-нотацією і Ω-нотацією?

O-нотація (Big O):

- **Описує верхню межу** асимптотичної поведінки.
- Використовується для характеристики найгіршого випадку.
- Гарантує, що алгоритм не буде повільнішим за вказану межу.
- Формально: $f(n)=O(g(n))f(n)=O(g(n))$ означає, що існують константи $c>0c>0$ і $n0n0$, такі що $f(n)\leq c\cdot g(n)f(n)\leq c\cdot g(n)$ для всіх $n\geq n0n\geq n0$.

Θ-нотація (Theta):

- **Описує точну межу** асимптотичної поведінки.
- Характеризує як верхню, так і нижню межу.
- Використовується, коли хочуть показати точний порядок зростання.

- Формально: $f(n)=\Theta(g(n))$ означає, що існують константи $c_1>0$, $c_2>0$ і n_0, такі що $c_1\cdot g(n)\le f(n)\le c_2\cdot g(n)$ для всіх $n\ge n_0$.

Ω-нотація (Omega):

- **Описує нижню межу** асимптотичної поведінки.
- Використовується для характеристики найкращого випадку.
- Гарантує, що алгоритм не буде швидшим за вказану межу.
- Формально: $f(n)=\Omega(g(n))$ означає, що існують константи $c>0$ і n_0, такі що $f(n)\ge c\cdot g(n)$ для всіх $n\ge n_0$.

Таким чином, основна різниця між цими нотаціями полягає в тому, що O-нотація визначає верхню межу, Ω-нотація – нижню межу, а Θ-нотація – точну межу асимптотичної складності.