

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ІМЕНІ МИХАЙЛА
ОСТРОГРАДСЬКОГО

Кафедра комп'ютерної інженерії та електроніки

ЗВІТ З ПРАКТИЧНОЇ РОБОТИ №1

з навчальної дисципліни

«Алгоритми та методи обчислень»

Тема «Асимптотична складність алгоритмів. О-нотація»

Студентка гр. КН-24-1 Бояринцова П. С.

Викладач Сидоренко В. М.

Тема роботи: Асимптотична складність алгоритмів. О-нотація

1.1 Постановка завдання.

Ознайомитися з теоретичними основами асимптотичного аналізу алгоритмів. Виконати індивідуальні завдання згідно з варіантом. Відповісти на контрольні питання.

1.2 Розв'язання задачі згідно зі своїм варіантом.

Задача 3: Дано функцію $f(n) = \log n + 2n^2 + 11$. Знайти асимптотичну складність у О-нотації.

Розв'язання:

Не зважаємо на константи: $f(n) = \log n + n^2$

За правилом спрощення, при великому n домінуючим членом є n^2 , оскільки він зростає швидше за $\log(n)$ (рис. 1). Отже, асимптотична складність: $f(n) = O(n^2)$.

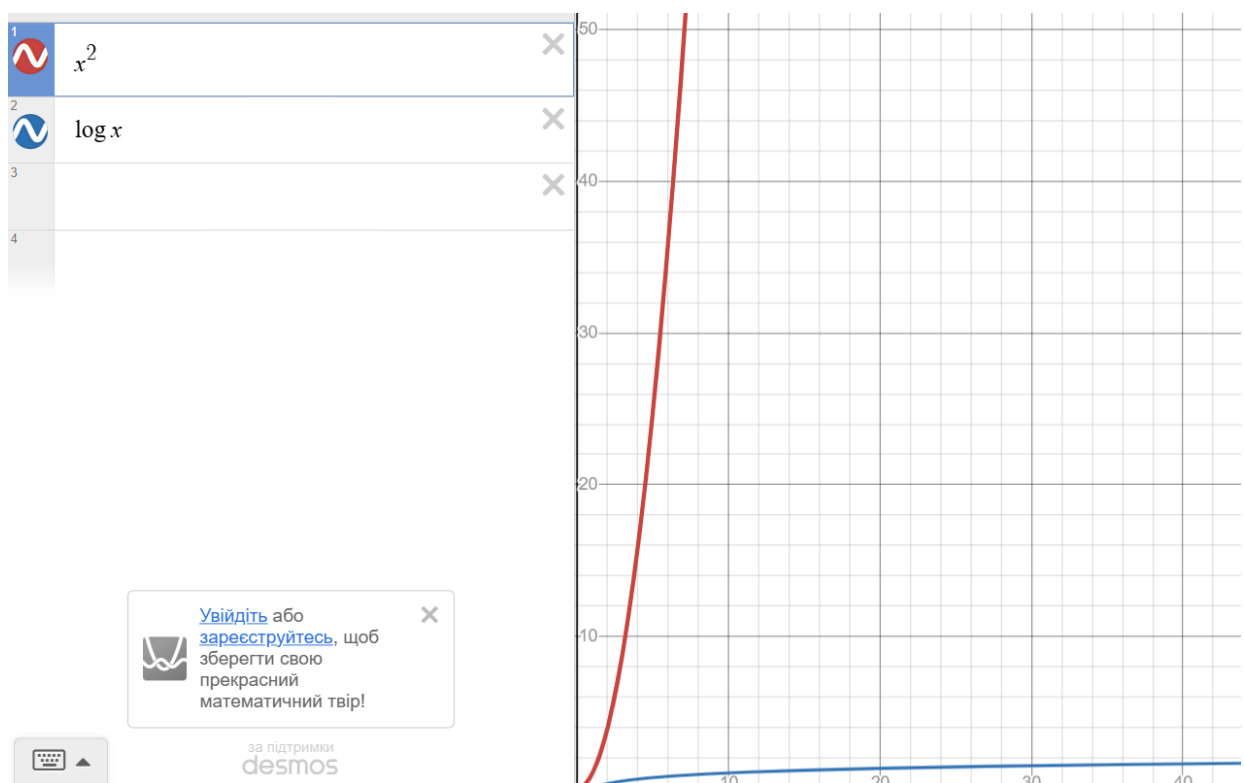


Рисунок 1 – Поведінка функцій

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

Розв'язання:

Згідно з визначенням О-нотації, потрібно знайти такі константи c і n_0 , що виконується: $2n^2 + 10n + 3 \leq cn^2$ для $\forall n \geq n_0$.

Маємо:

$$2n^2 + 10n + 3 \leq 2n^2 + 10n^2 + 3n^2, \text{ для } n \geq 1.$$

$$2n^2 + 10n + 3 \leq 15n^2$$

Отже, можна взяти $c = 15$, $n_0 = 1$. Це доводить, що $f(n) = O(n^2)$.

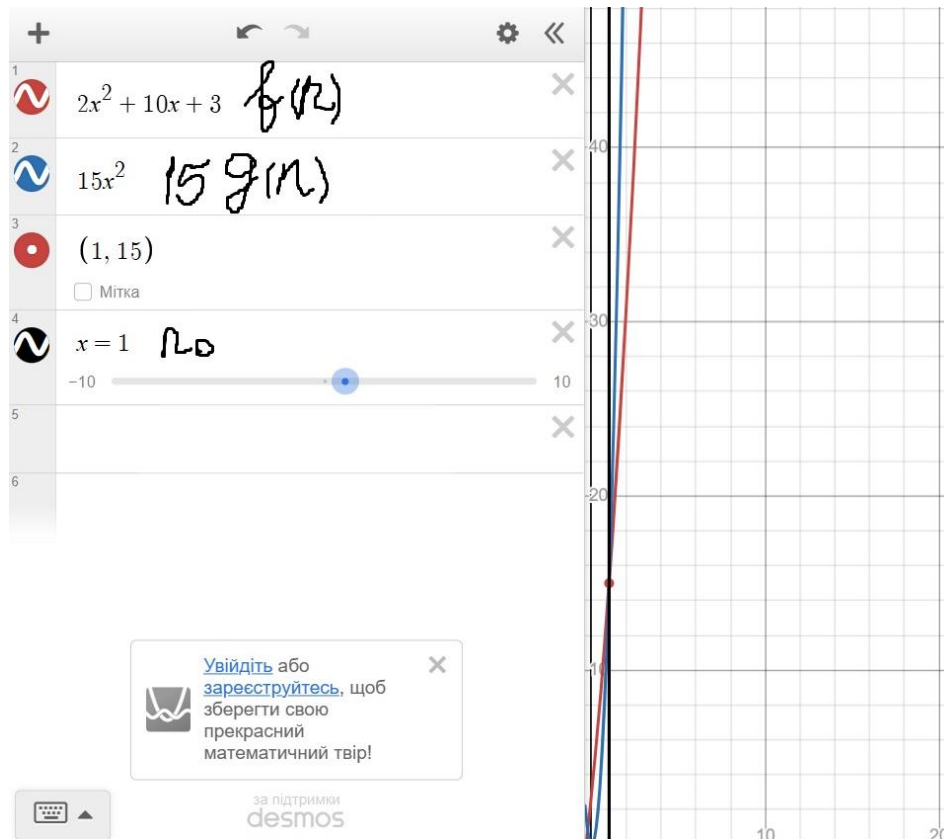


Рисунок 2 – Графік до задачі 8

1.3 Отримані результати.

Визначено асимптотичну складність заданих функцій.

Доведено правильність обчислень згідно з визначенням О-нотації.

1.4 Відповіді на контрольні питання.

1. Що таке асимптотична складність алгоритму?

Асимптотична складність алгоритму – це міра зростання часу виконання або використання пам'яті алгоритмом при збільшенні вхідних даних.

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

О-нотація використовується для оцінки верхньої межі зростання функції. Вона визначається як множина функцій, які не зростають швидше за певну граничну функцію з точністю до константи.

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

Залишаємо тільки найшвидше зростаючий член.

Ігноруємо константні множники.

Використовуємо визначення О-нотації для формального доведення.

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

$O(1)$ – алгоритм виконується за сталий час, незалежно від розміру вхідних даних.

$O(n)$ – час виконання алгоритму зростає лінійно зі збільшенням розміру вхідних даних.

$O(n^2)$ – час виконання алгоритму зростає квадратично.

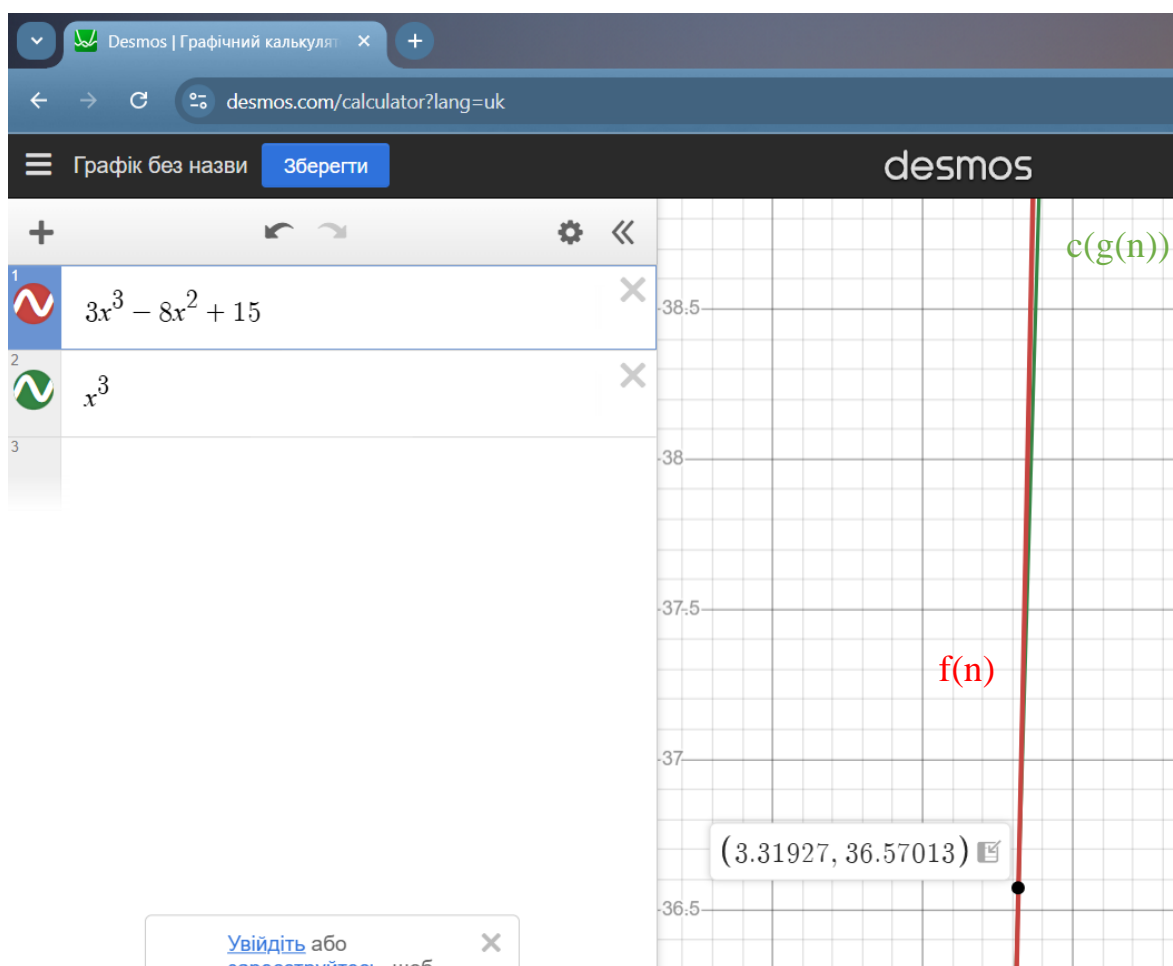
5. Яким чином визначити асимптотичну складність алгоритму за його кодом або математичним виразом?

Аналізувати вкладені цикли та рекурсію.

Використовувати правила спрощення О-нотації.

Виділяти найшвидше зростаючий член у виразі складності.

3. $f(n) = 3n^3 - 8n^2 + 15$
 $g(n) = n^3$
 Довести, що $f(n) = \Omega(g(n))$
 $f(n) = n^3 + (2n^3 - 8n^2) + 15$
 $2n^3 - 8n^2 > 0$ для $\forall n > n_0 = 4$
 $3n^3 - 8n^2 + 15 = n^3 + (2n^3 - 8n^2) + 15$
 $\geq n^3 + 15$
 $\geq 1 \cdot n^3$
 $C = 1; n_0 = 4$



8. $f(n) = n^4 + 2n^3 - 5n^2 + 8$
 $g(n) = n^4$
Показати $f(n) = O(g(n))$

$$\lim_{n \rightarrow \infty} \frac{n^4 + 2n^3 - 5n^2 + 8}{n^4} = \lim_{n \rightarrow \infty} \frac{n^4}{n^4} = 1$$

$1 > 0$ і $1 \neq \infty$

