Алгоритми та структури даних
Модуль 1: Структури даних
Лекція 2. Основи аналізу ефективності алгоритмів.

Кудін Олексій Володимирович
avk256@gmail.com

План

1. Основи аналізу алгоритмів.
2. Асимптотичні позначення й основні класи ефективності.
3. Емпіричний аналіз алгоритмів.

Питання по попередній лекції

- *Дайте інтуїтивне визначення алгоритма.*
- *Що таке евристичний алгоритм?*
- *Які існують способи запису алгоритмів?*
- *Що таке машина Тюринга?*
- *Сформулюйте тезис Тюригна.*
- *Що таке нормальний алгоритм алгоритм Маркова?*

Ефективність алгоритма

За якими параметрами можна виміряти ефективність алгоритма?

Ефективність алгоритма

- Ч**асова ефективність** (time efficiency) є індикатором швидкості роботи алгоритму.
- **Просторова ефективність** (space efficiency) показує, скільки додаткової оперативної пам'яті потрібно для роботи алгоритму.

Оцінка розміру вхідних даних

Яким чином можна оцінити розмір вхідних даних для різних типів задач?

Час виконання алгоритма

Час виконання програми залежить від:

- швидкодії конкретного комп'ютера;
- старанності реалізації алгоритму у вигляді програми;
- типу компілятора, використаного для генерації машинного коду;
- точності хронометрування реального часу виконання програми.

Час виконання алгоритма

Основна (базова) операція (basic operation) алгоритма – це операція, яка виконується на кожній ітерації алгоритма.

Зазвичай знаходиться у найбільш вкладеному циклі.

Час виконання алгоритма

c_{op} — час виконання основної операції алгоритму на конкретному комп'ютері, а $C(n)$ — кількість разів, які ця операція повинна бути виконана при роботі алгоритму.

Тоді час виконання програмної реалізації цього алгоритму на даному комп'ютері:

$$T(n) \approx c_{op} C(n)$$

Порядок росту функції

n	$\log_2 n$	n	$n \log_2 n$	n^2	n^3	2^n	$n!$
10	3.3	10	$3.3 \cdot 10^1$	10^2	10^3	10^3	$3.6 \cdot 10^6$
10^2	6.6	10^2	$6.6 \cdot 10^2$	10^4	10^6	$1.3 \cdot 10^{30}$	$9.3 \cdot 10^{157}$
10^3	10	10^3	$1.0 \cdot 10^4$	10^6	10^9		
10^4	13	10^4	$1.3 \cdot 10^5$	10^8	10^{12}		
10^5	17	10^5	$1.7 \cdot 10^6$	10^{10}	10^{15}		
10^6	20	10^6	$2.0 \cdot 10^7$	10^{12}	10^{18}		

Асимптотичні визначення

Говорять, що функція $t(n)$ належить множині $O(g(n))$, що записується як $t(n) \in O(g(n))$, якщо для всіх великих значень n функція $t(n)$ обмежена зверху деякою константою, помноженої на значення функції $g(n)$, тобто якщо існує позитивна константа c й ненегативне ціле число n_0 таке, що $t(n) \le cg(n), \forall n \ge n_0$.

Асимптотичні визначення

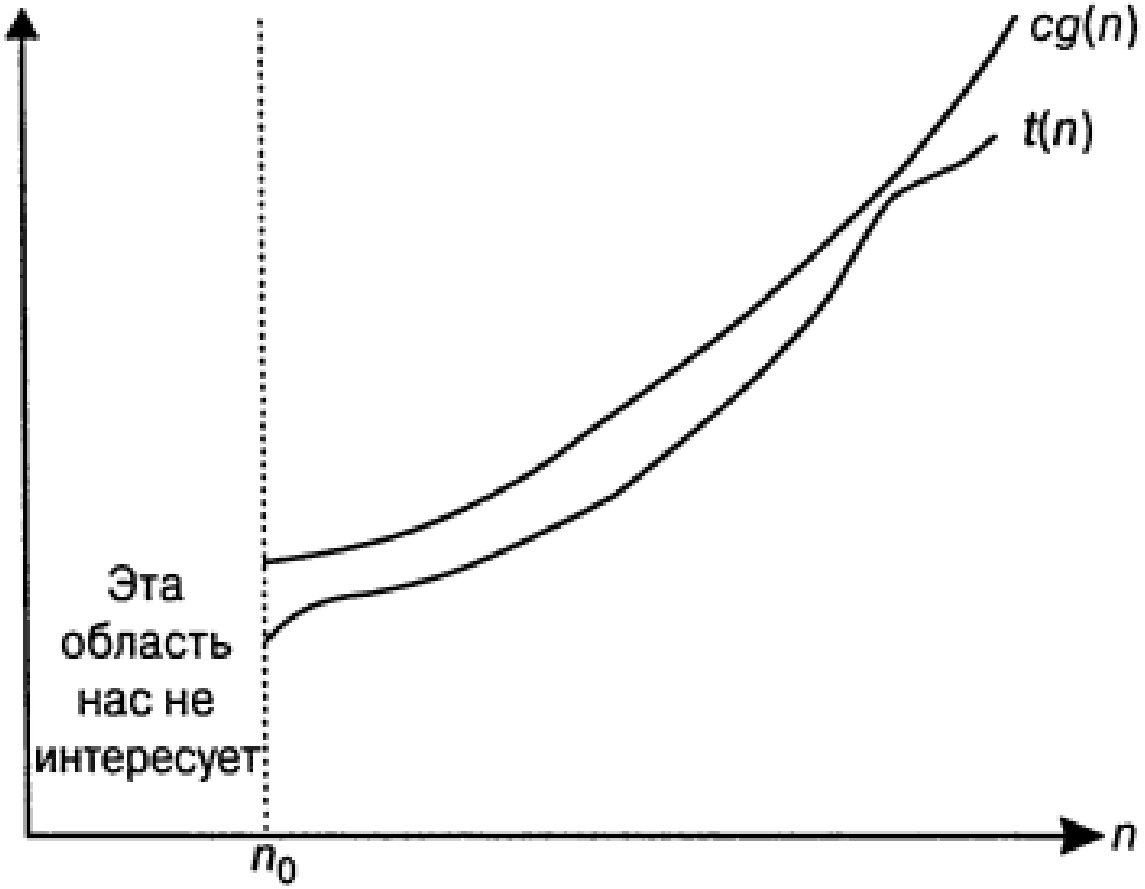

Рис. 2.1. Иллюстрация O-обозначения: $t(n) \in O(g(n))$

Асимптотичні визначення

Говорять, що функція $t(n)$ належить множині $\Omega(g(n))$, що записується як $t(n) \in \Omega(g(n))$, якщо для всіх великих значень n функція $t(n)$ обмежена знизу деякою константою, помноженої на значення функції $g(n)$, тобто якщо існує позитивна константа c й ненегативне ціле число n_0 таке, що $t(n) \geq cg(n), \forall n \geq n_0$.

Асимптотичні визначення

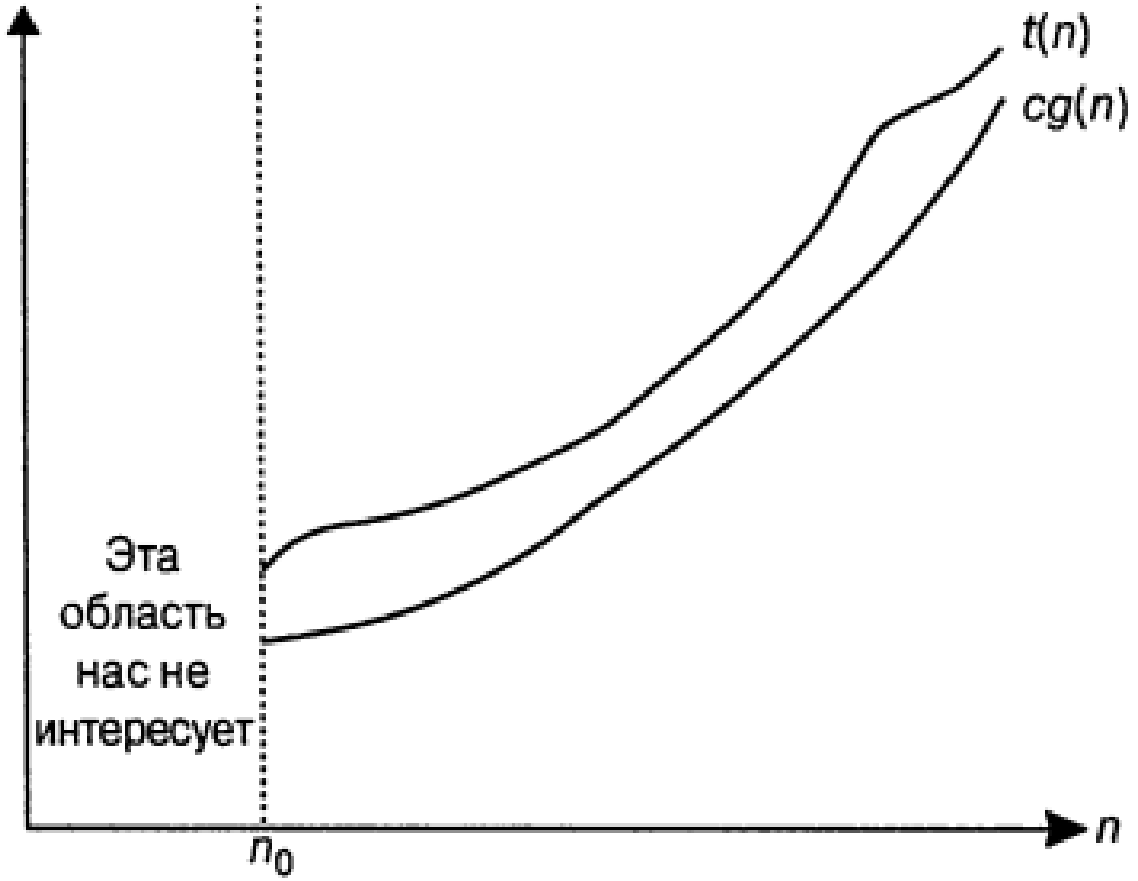

Рис. 2.2. Иллюстрация Ω-обозначения: $t(n) \in \Omega(g(n))$

Асимптотичні визначення

Говорять, що функція $t(n)$ належить множині $\Theta(g(n))$, що записується як $t(n) \in \Theta(g(n))$, якщо для всіх великих значень n функція $t(n)$ обмежена знизу й зверху деякими константами, помноженими на значення функції $g(n)$, тобто якщо існують позитивні константи c_1, c_2 й ненегативне ціле число n_0 таке, що $c_2 g(n) \le t(n) \le c_1 g(n), \forall n \ge n_0$.

Асимптотичні визначення

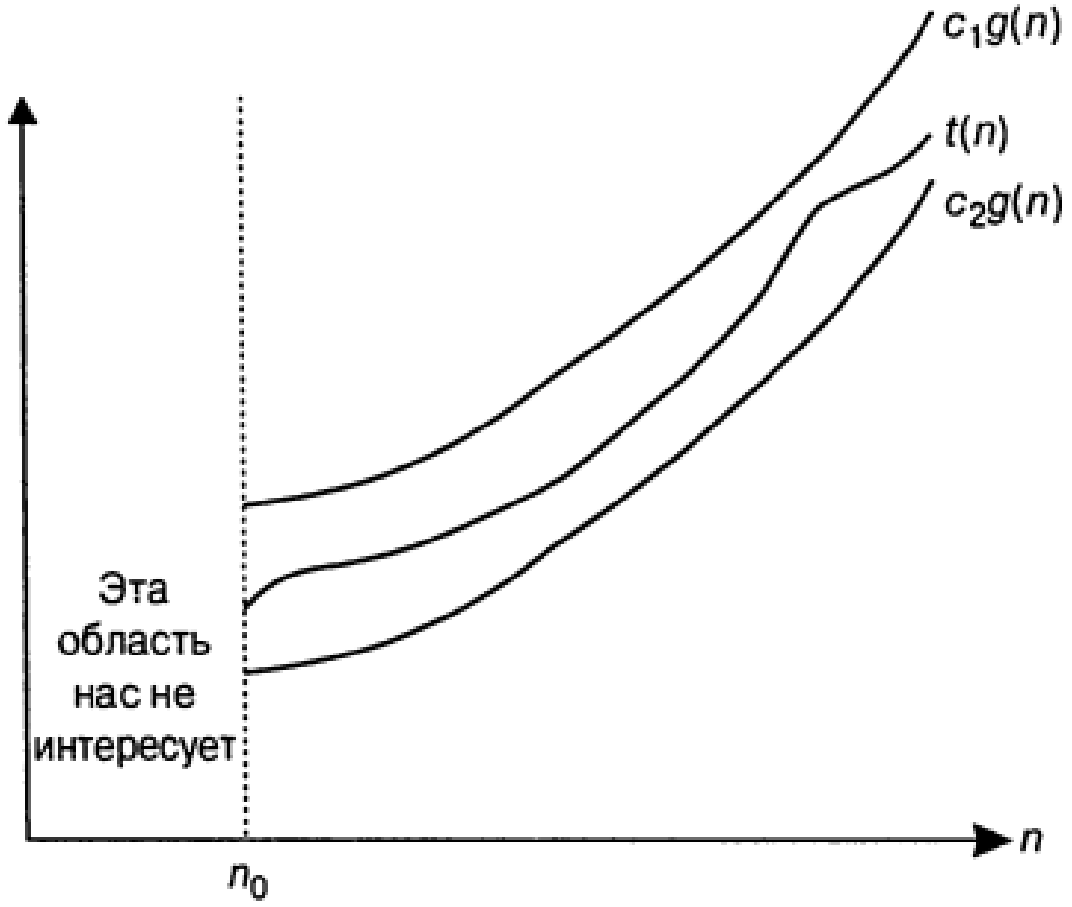

Рис. 2.3. Иллюстрация Θ-обозначения: $t(n) \in \Theta(g(n))$

Теорема 1. Якщо $t_1 \in O(g_1(n))$ й $t_2 \in O(g_2(n))$ то $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$. Аналогічні твердження справедливі також для Ω, Θ.

$$\lim_{n\to\infty} \frac{t(n)}{g(n)} = \begin{cases} 0 & \text{если } t(n) \text{ имеет меньший порядок роста, чем } g(n) \\ c & \text{если } t(n) \text{ имеет тот же порядок роста, чем } g(n) \\ \infty & \text{если } t(n) \text{ имеет больший порядок роста, чем } g(n). \end{cases}$$

Класи ефективності

$\log n$	*Логарифмическая*	Обычно такая зависимость появляется в результате сокращения размера задачи на постоянное значение на каждом шаге итерации алгоритма (см. раздел 5.5). Обратите внимание, что в логарифмическом алгоритме невозможна работа со всеми входными данными (и даже с их некоторой фиксированной частью). В этом случае время выполнения любого алгоритма будет находиться по меньшей мере в линейной зависимости от размера входных данных
n	*Линейная*	К этому классу относятся алгоритмы, выполняющие сканирование списка, состоящего из n элементов, например алгоритм поиска методом последовательного перебора
$n \log n$	*n-log-n*	К этому классу относятся большое количество алгоритмов декомпозиции (см. главу 4), таких как алгоритмы сортировки слиянием и быстрой сортировки

Класи ефективності

n^2	*Квадратичная*	Как правило, подобная зависимость характеризует эффективность алгоритмов, содержащих два встроенных цикла (см. следующий раздел). В качестве типичных примеров достаточно назвать простой алгоритм сортировки и целый ряд операций, выполняемых над матрицами размером $n \times n$
n^3	*Кубическая*	Как правило, подобная зависимость характеризует эффективность алгоритмов, содержащих три встроенных цикла (см. следующий раздел). К этому классу относятся несколько довольно сложных алгоритмов линейной алгебры
2^n	*Экспоненциальная*	Данная зависимость типична для алгоритмов, выполняющих обработку всех подмножеств некоторого множества, состоящего из n элементов. Часто термин "экспоненциальный" используется в широком смысле и означает очень высокие порядки роста, т.е. включает более быстрые по сравнению с экспонентой порядки роста

Ефективність алгоритма для різних вхідних даних

Розглядають такі види ефективності в залежності від структури та розміру вхідних даних:

- **Ефективність в найгіршому випадку** (worst-case efficiency).
- **Ефективність в найкращому випадку** (best-case efficiency).
- **Ефективність в середнього випадку** (average-case efficiency).

Загальний план емпіричного аналізу ефективності алгоритму

- 1. З'ясування мети майбутнього експерименту.
- 2. Визначення вимірюваної метрики M и одиниць виміру (кількість операцій або час роботи).
- 3. Визначення характеристик вхідних даних (їхній діапазон, розмір і т.д.).
- 4. Створення програми, що реалізує алгоритм (або алгоритми), для проведення експерименту.
- 5. Генерація зразка вхідних даних.
- 6. Виконання алгоритму (або алгоритмів) над зразком вхідних даних і запис спостережуваних даних.
- 7. Аналіз отриманих даних.

Емпіричний аналіз

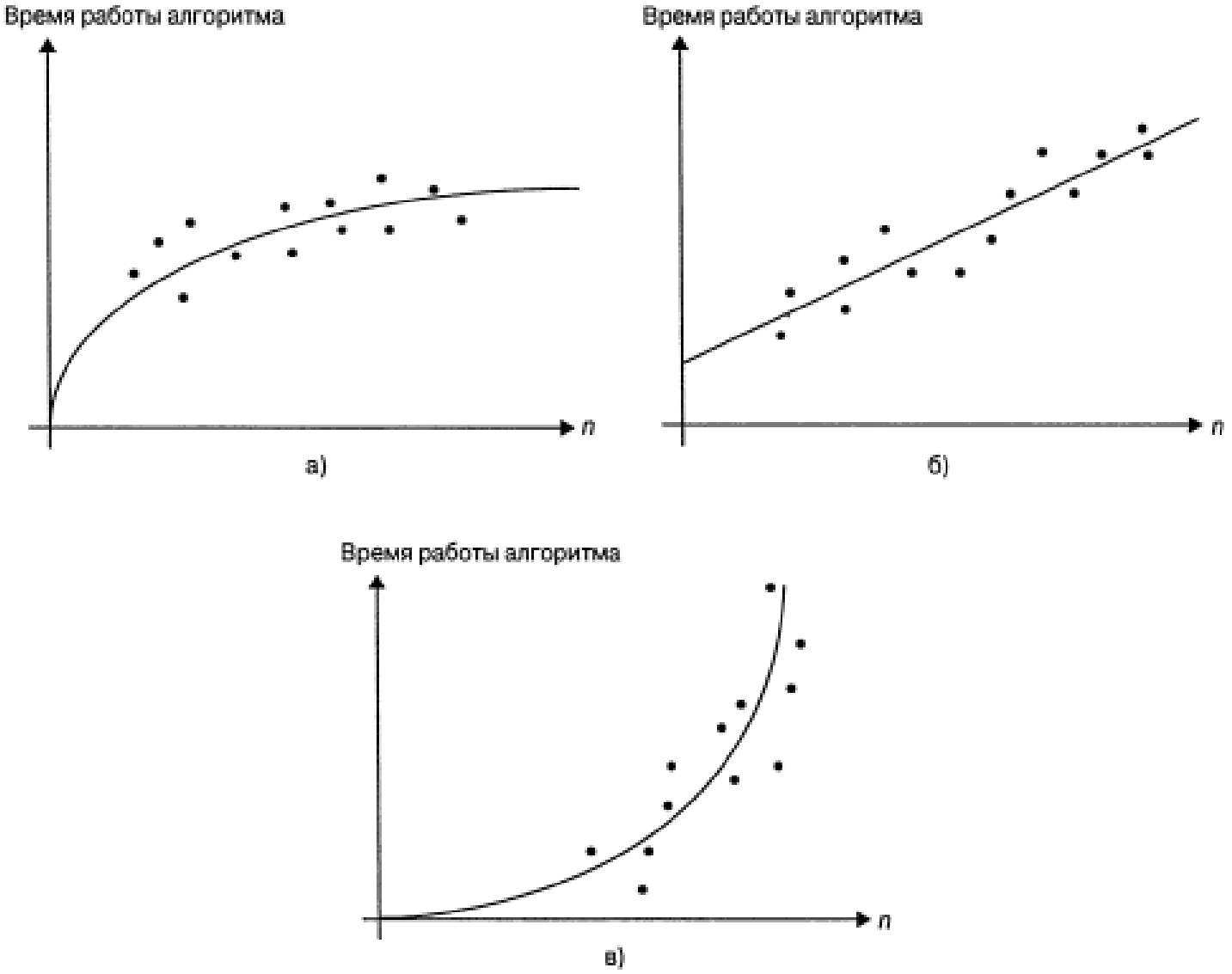

Рис. 2.7. Типичные графики функций: а) логарифмической, б) линейной, в) выпуклой вниз функции

Резюме

Резюме (що встигли за лекцію)

- Розглянули основні параметри ефективності.
- Дали визначення асимптотичним позначенням.
- Розглянули основні класи ефективності.
- Розглянули план емпіричного аналізу.

Завдання до наступної лекції

- Навчитися оперувати з асимптотичними позначеннями.