

## Асимптотическая оценка алгоритмов

- $t(n)$  — функциональная зависимость между объемом входных данных и кол-вом этих операций

$t(n)$  — неизвестная функция, где  $n \rightarrow \infty$

### $O$ — нотация

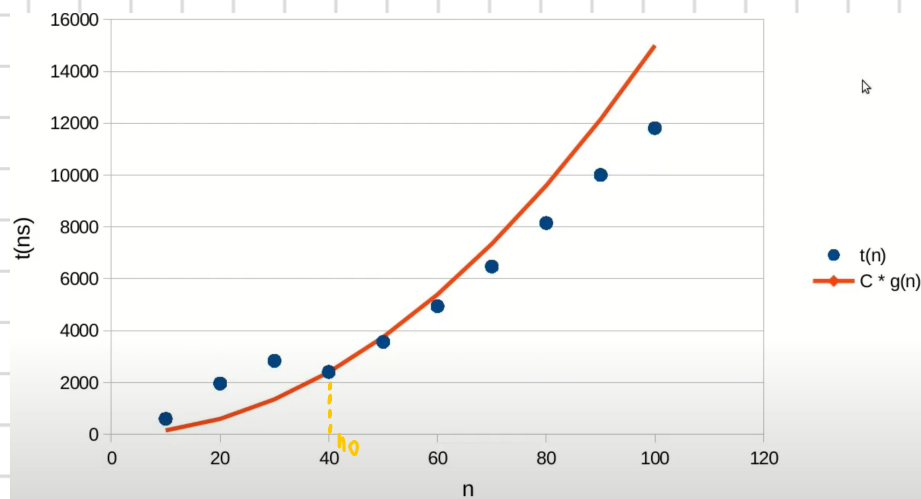
$t(n) = O(g(n))$  означает, что  $t$  — функция, которая растет не быстрее чем  $g \cdot C$ , где  $C$  — константа

$$\exists C > 0, \exists n_0 : \forall n > n_0 : |t(n)| \leq |C \cdot g(n)|$$

Оценка  $O$  представляет собой верхнюю

асимптотическую оценку

трудоемкости алгоритма

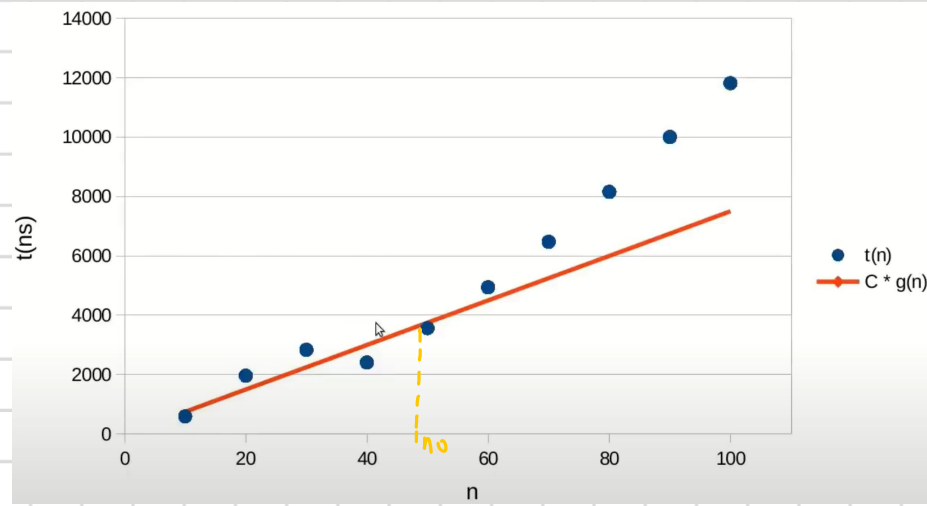


### $\Omega$ — нотация (омега)

$t(n) = \Omega(g(n))$  означает, что  $t$  — ф-ия которая растет не медленнее чем  $g(n) \cdot C$ ,  $C$  — константа

$$\exists C > 0, \exists n_0 : \forall n > n_0 : |C \cdot g(n)| \leq |t(n)|$$

оценка  $\Omega$  дает  
нижнюю асимптотическую  
оценку роста ф-ии  $t(n)$

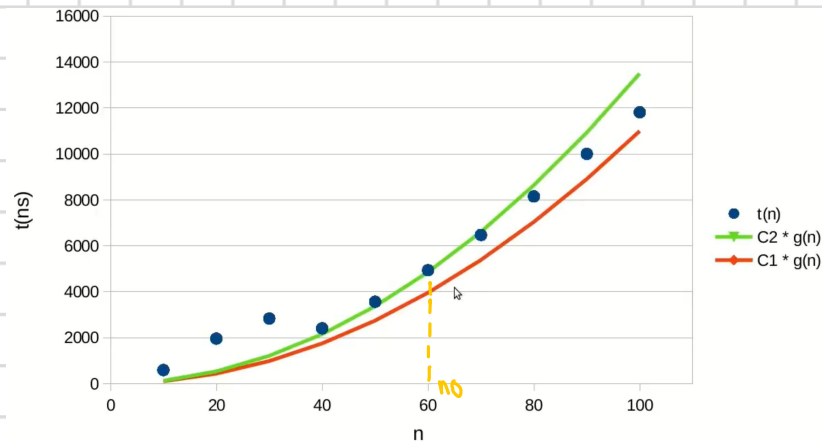


$\Theta$  — тета (тета)

$t(n) = \Theta(g(n))$  означает, что  $g(n)$  является асимптотически  
точной оценкой  $t(n)$

$$\exists C_1 > 0, C_2 > 0 \exists n_0 : \forall n > n_0 : |C_1 \cdot g(n)| \leq |t(n)| \leq |C_2 \cdot g(n)|$$

$\Theta$  дает одновременно верхнюю  
и нижнюю границу оценки  
роста функции



## Асимптотические сложности

Сложность	Описание	Пример
$O(1)$	Константная. Не зависит от объема данных.	Получение элемента массива по индексу
$O(\ln(n))$	Логарифмическая	Бинарный поиск
$O(n)$	Линейная.	Поиск элемента не сортированном массиве
$O(n \ln(n))$	Линеаритмичная.	Быстрая сортировка
$O(n^2)$	Квадратичная.	Сортировка выбором
$O(n^3)$	Кубическая.	Перемножение матриц
$O(n^k)$	Полиномиальная.	Алгоритм Кармаркара
$O(C^n)$	Экспоненциальная.	Задачи полного перебора графа
$O(n!)$	Факториальная.	Комбинаторные алгоритмы

# Графики асимптот-их сложн.

