

# **Основы анализа алгоритмов.**

**Асимптотический анализ сложности алгоритмов.**

**Лекция 2.**

Доцент КВТ Лапина О.Н.

$\pi$

# O-символика.

- › 1. Функция  $T(n)$  имеет порядок  $O(f(n))$  если существуют константы  $c > 0$  такие, что для всех  $n > n_0$  выполняется неравенство  $T(n) \leq cf(n).$
- › В этом случае существует константа  $C > 0$  такая, что  $\lim_{n \rightarrow \infty} \frac{T(n)}{f(n)} = C$
- › Т.е. отношение  $|T(n)| / |f(n)|$  ограничено сверху.  
1
- › Для алгоритмов, функция сложности которых имеет порядок  $O(f(n))$
- › говорят, что она имеют порядок (или степень) роста  $f(n)$
- › Подразумевается, что  $f(n)$  является **верхней границей скорости роста  $T(n)$** .

# O-символика.

- › Главная характеристика алгоритма – **скорость решения задачи** (временная эффективность).
- › Вторая важная характеристика – **сложность реализации**.
  
- › Для анализа алгоритмов используют теоретические методы, основным из которых является **метод асимптотической оценки временной эффективности**. Суть его заключается в установлении функциональной зависимости числа выполняемых алгориттом действий от объема исходных данных, в предельном случае больших объемов этих данных.

# О-символика.

- › 2. Функция  $T(n)$  является **о-малым** от  $f(n)$  при  $n \rightarrow \infty$ 
  - › если для любого  $\varepsilon > 0$  существует  $n_0 > 0$ , что для всех  $n > n_0$  выполняется неравенство  $T(n) \leq \varepsilon f(n)$ .
  - › В этом случае  $\lim_{n \rightarrow \infty} \frac{T(n)}{f(n)} = 0$
  - › Т.е. отношение  $|T(n)| / |f(n)|$  стремиться к нулю.
- › 3. Чтобы указать нижнюю границу скорости роста  $T(n)$  используется  $\Omega(g(n))$
- › Это означает, что существует такая константа  $c > 0$ , что для бесконечного числа значений  $n$  выполняется неравенство:

$$T(n) \geq c g(n).$$

# O-символика.

- › Пример 1. Функция  $T(n) = n^3 + 2n^2$  есть  $\Omega(n^2)$
- › Так как  $T(n) \geq cn^2$  при  $c=1$  для  $n=0, 1, 2, 3, \dots$
- › Пример 2.

$$T(n) = \begin{cases} n, & n - \text{нечетны} \\ n^2/100, & n - \text{четн} \end{cases}$$

$$T(n) = \Omega(n^2)$$

- › Если  $c = 1/100$        $n=2, 4, \dots$  четное       $T(n) \geq cn^2$
- › Из правила суммы следует:  $O(n^3 + n) = O(n^3)$

# O-символика.

› Таблица обозначений:

$T(n) = O(f(n))$	$T(n)$ ограничена сверху функцией $f(n)$	$\exists c > 0$ и $n_0 > 0$ , что для всех $n > n_0, T(n) \leq cf(n)$ .
$T(n) = o(f(n))$	$f(n)$ доминирует над $T(n)$ асимптотически	$\forall c > 0$ , $\exists n_0 > 0$ , что для всех $n > n_0, T(n) \leq cf(n)$ .
$T(n) = \Omega(f(n))$	$T(n)$ ограничена снизу функцией $f(n)$	$\exists c > 0$ и $n_0 > 0$ , что для всех $n > n_0, T(n) \geq cf(n)$ .
$T(n) = \omega(f(n))$	$T(n)$ доминирует над $f(n)$ асимптотически	$\forall c > 0$ , $\exists n_0 > 0$ , что для всех $n > n_0, T(n) \geq cf(n)$ .
$T(n) = \Theta(f(n))$	$T(n)$ ограничена сверху и снизу функцией $f(n)$	$\exists c > 0$ , $c' > 0$ и $n_0 > 0$ , что для всех $n > n_0, cf(n) \leq T(n) \leq c'f(n)$ .
$T(n) \leftrightarrow f(n)$	$T(n)$ эквивалентна $f(n)$ асимптотически	$\lim_{n \rightarrow \infty} \frac{T(n)}{f(n)} = 1$

# O-символика.

- › Используя O-символику для асимптотической оценки скорости роста функций сложности, мы пренебрегаем константами пропорциональности.
- › Пусть у нас имеется 2 программы:  $100n^2 = O(n^2)$  и  $5n^3 = O(n^3)$
- › Какая из этих 2-х программ будет предпочтительней?
- › **Ответ на этот вопрос зависит:**
  - › 1) от размера входных данных программ.
  - › 2) если время работы 2-х программ примерно одинаково, предпочтение следует отдать программе с меньшей степенью роста, как более надежной программе.

$\pi$

# O-символика.

› Рассмотрим функций сложности алгоритма для 4-х программ

Время выполнения $T(n)$	Макс. размер задачи для 1000 сек.	Макс. размер задачи для 10 000 сек.	Увеличение макс. размера задачи
$100n$	10	100	10,0
$5n^2$	14	45	3,2
$n^3 / 2$	12	27	2,3
$2^n$	10	13	1,3

$\pi$ 

# О-символика.

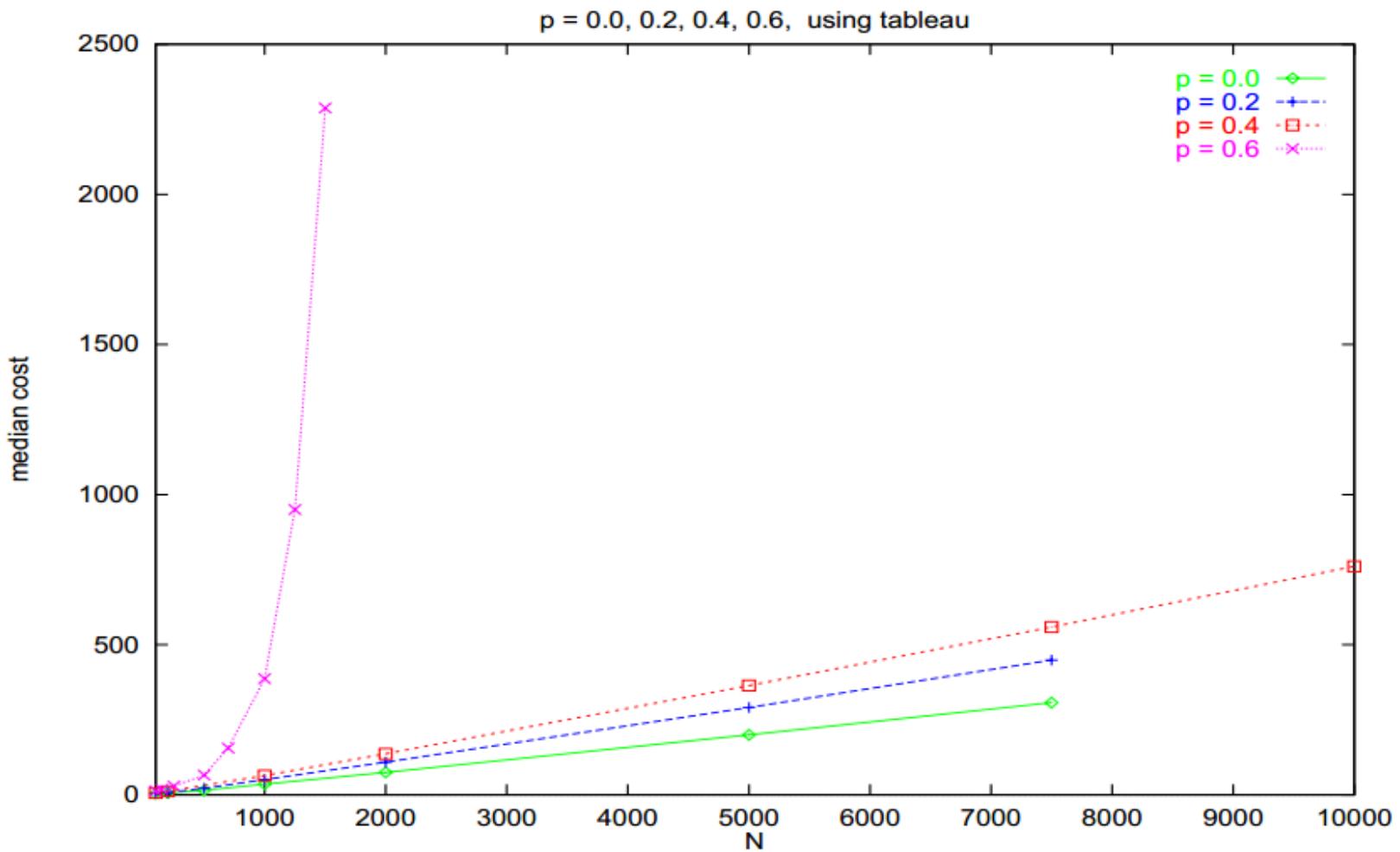


Fig. 8a: Median computational cost, linear scale.