

Понятие сложности алгоритма

Проблема:

*Ресурсы ЭВМ (ОП и время процессора) ограничены, следует выбирать из эквивалентных алгоритмов наиболее **эффективный**.*

*Для оценки эффективности введено понятие **сложности алгоритма**.*

От чего зависит сложность?

- *времени, затраченного на выполнение алгоритма*
- *объема памяти, требуемой для хранения исходных данных задачи.*

Временная сложность алгоритма

*- это время **T**, необходимое для его выполнения в зависимости от исходных данных.*

$$**T = k \cdot t**, где$$

k – кол-во вычислительных действий,
t – время выполнения одного действия.

Объемная сложность алгоритма

*Зависит от объема памяти,
используемой для размещения исходных
данных и результатов программы.*

$T(n)$ – зависимость времени от
объема входных данных.

n – это размерность задачи (для
линейного массива – размер массива).

Поведение **T** при увеличении **n**
называется теоретической
сложностью – **$O(f(n))$** .

Сравнительная оценка сложности алгоритмов

| Сложность $O(f(n))$ | Тип зависимости | Значение при $n=2^{10}=$ $=1024$ |
|------------------------|-------------------------|--|
| $O(n)$ | <i>Линейная</i> | 1024 |
| $O(n \cdot \log_2 n)$ | <i>Логарифмическая</i> | 10240 |
| $O(n^2)$ | <i>Полиномиальная</i> | $\approx 10^6$ |
| $O(n^3)$ | | $\approx 10^9$ |
| $O(2^n)$ | <i>Экспоненциальная</i> | $\approx 10^{300}$ |

Задача

Дано: два алгоритма A_1 и A_2 , решающих одну и ту же задачу размерности $n=10^6$.

A_1 имеет сложность $O_1(n^2)$ и выполняется на суперкомпьютере с быстродействием 10^8 оп/с;

A_2 имеет сложность $O_2(n \cdot \log_2 n)$ и выполняется на обычном компьютере с быстродействием 10^6 оп/с.

Требуется: найти время решения

задачи $t_1 - ?$, $t_2 - ?$

Решение

$$t_1 = 10^{12} / 10^8 = 10^4 \text{ с} \approx 2,8 \text{ ч}$$

$$t_2 = 10^6 \cdot \log_2 10^6 / 10^6 = 6 \cdot \log_2 10 \approx 20 \text{ с}$$

Вывод: *Разработка эффективных алгоритмов не менее важна, чем разработка быстрой электроники!*