Оценка сложности алгоритмов

Лекция 1.

Сложность алгоритма: понятие, виды сложности. Классы сложности.

Алгоритмы

- Вспомним, что такое «алгоритм».
- Под «алгоритмом» обычно понимают четко определенную последовательность действий, приводящую через конечное число шагов к результату — решению задачи, для которой разработан алгоритм.

Алгоритмы

Основные свойства, присущие любому алгоритму:

- массовость — алгоритм предназначен для решения задачи с некоторым множеством допустимых входных данных;
- конечность — алгоритм должен завершаться за конечное число шагов.

Алгоритмы

- Не для любой задачи существует алгоритм решения. Существуют *алгоритмически неразрешимые* задачи.
- Но даже если алгоритм существует, он может оказаться неприменимым на практике из-за высокой *сложности*.

Сложность алгоритма

- *Сложность алгоритма* – это количественная характеристика ресурсов, необходимых алгоритму для работы (успешного решения задачи).
- Основные ресурсы:
  - **время** (*временнáя сложность*) и
  - **объем** памяти (*ёмкостная сложность*).
- Наиболее важной характеристикой является *время*.

Сложность алгоритма

- Сложность задачи может быть разной для разных входных данных (*экземпляров* задачи).
- Различают *сложность в худшем случае* и *сложность в среднем*.
- В теории сложности чаще оперируют понятием сложности в худшем случае.

Сложность алгоритма

Обычно оценивают порядок роста сложности при $n \to \infty$: $T = O(f(n))$.

Почему?

- Фактическая сложность (время работы в секундах) зависит не только от алгоритма, но и от скорости работы компьютера.
- Именно порядок роста сложности ограничивает размер решаемых задач.

Сложность алгоритма

n (размер задачи)	$T=O(n)$	$T=O(2^n)$
50	1 сек	1 сек
51	1,02 сек	2 сек
60	1,2 сек	17 мин
70	1,4 сек	12 суток
80	1,6 сек	34 года
90	1,8 сек	~35 тыс.лет

Сложность задачи

- Нас интересует не только сложность конкретного алгоритма, решающего задачу, но и сложность *задачи* в целом.
- ***Сложность задачи*** естественно определить как сложность самого эффективного алгоритма, решающего эту задачу.

Сложность задачи

- К сожалению, это невозможно!
- Доказано, что есть задачи, для которых *не существует* самого быстрого алгоритма, потому что любой алгоритм для такой задачи можно «ускорить», построив более быстрый алгоритм, решающий эту задачу.

Сложность задачи

Теорема Блюма об ускорении (упрощенный вариант).

Существует такая алгоритмически разрешимая задача Z, что любой алгоритм A, решающий задачу Z, можно ускорить следующим образом: существует другой алгоритм A', также решающий Z и такой, что $T_{A'}(n) \le \log T_A(n)$ *для почти всех n.*

Классы сложности

- Выход: вместо «сложности задачи» рассматривать <u>*классы сложности*</u>.
- Определение. *Пусть f*(*n*) — *некоторая функция, отображающая N в N.* Класс сложности *C*(*f*(*n*)) — *это множество всех задач, для которых существует хотя бы один алгоритм, сложность которого не превышает O*(*f*(*n*)).

Классы сложности

- Это определение является неполным.
- В полном определении необходимо уточнить:
  - что мы понимаем под «алгоритмом»;
  - какая сложность (временная, емкостная или какая-нибудь еще) нас интересует.
- К этим уточнениям мы приступим на следующей лекции…

Рекомендуемая литература

- Адигеев М.Г. Введение в теорию сложности – Методические указания. — Ростов-н/Д, 2004 г.
- Кузюрин Н.Н. Курс лекций «Сложность комбинаторных алгоритмов»: http://discopal.ispras.ru/ru.lectures.htm
- Разборов А.А. О сложности вычислений — Математическое просвещение — сер. 3, вып. 3, 1991 г.

  http://www.mccme.ru/free-books/matpros/i4127141.pdf.zip