

Дисциплина
«РАЗРАБОТКА ЭФФЕКТИВНЫХ АЛГОРИТМОВ»

Лекция 1

**ИСТОРИЧЕСКИЙ ОЧЕРК И АЛГОРИТМИЧЕСКОЕ
ОБЕСПЕЧЕНИЕ ПРОГРАММНЫХ РАЗРАБОТОК**

Лектор: Михаил Васильевич Ульянов,

**д.т.н., профессор, ведущий научный сотрудник ИПУ РАН,
профессор ВМК МГУ, МГТУ им. Н.Э. Баумана, ДПИ ФКН ВШЭ**

muljanov@mail.ru, 8 916 589 94 04

ИСТОИЧЕСКИЙ ОЧЕРК

Несомненные успехи математики и физики, начиная с середины XVIII века, и на протяжении XIX века (описание наблюдаемого мира дифференциальными уравнениями в частных производных — Эйлер, Лагранж, Лаплас, Навье-Стокс и т.д.), привели в начале XX к некоторой «эйфории» о мощности науки. Еще немного и получится математически описать все процессы и, более того, построить красивое здание самой математики. Отражением этой эйфории стало выступление Д. Гильберта на II-ом Международном математическом конгрессе в Париже в 1900 году.

1.1900 год. II Международный математический конгресс в Париже. Давид Гильберт — список из 23 проблем математики — «Проблемы Гильберта». Математики столкнулись с существенными трудностями в решении этих проблем. Некоторые из них не решены до сих пор.

2.1930 год. Теорема К. Гёделя о неполноте и вторая теорема Гёделя — две теоремы математической логики о принципиальных ограничениях формальной арифметики и, как следствие, всякой формальной системы, в которой можно определить основные арифметические понятия: натуральные числа, 0, 1, сложение и умножение. Первая теорема утверждает, что если формальная арифметика непротиворечива, то в ней *существует невыводимая и неопровержимая формула*.

Вторая теорема утверждает, что если формальная арифметика непротиворечива, то в ней невыводима некоторая формула, содержательно утверждающая непротиворечивость этой арифметики. Обе эти теоремы были доказаны Куртом Гёделем в 1930 году и имеют непосредственное отношение ко второй проблеме из знаменитого списка Гильберта.

1. Современная теория алгоритмов

Начало теории — 1936 год. Статья Алана Тьюринга — «О проблеме разрешимости» (в рамках второй проблемы Гильберта) и статья Э. Л. Поста положили начало современной теории алгоритмов. Основные задачи теории, рассматривающей **алгоритм** как объект исследования: формализация понятия алгоритма (машина Тьюринга, машина Поста, частично-рекурсивные функции Чёрча-Клини, нормальные алгорифмы Маркова) и изучение основных свойств алгоритмов и особенностей. (Машина Поста — тема следующей лекции.) Основным результатом этой теории в рамках решения проблем Гильберта — доказательство существования **алгоритмически неразрешимых задач**. В 1970 году Юрием Матиясевичем была доказана алгоритмическая неразрешимость проблемы нахождения универсального метода определения разрешимости произвольного алгебраического диофантова уравнения (10-ая проблема).

2. Теория сложности вычислений

Появление в 1964 году серийно выпускаемой вычислительной машины IBM 360 привело к существенному расширению круга решаемых на ЭВМ задач. И у математиков и программистов появилось понимание того, что сами задачи могут быть классифицированы по сложности.

Основы теории были заложены Эдмондсом и Карпом в середине 1960-х, собственно стройной теория стала в 1971 году, после теоремы Стивена Кука.

Было введено понятие сложности и предложены классификации задач, использующие асимптотические оценки зависимости условного «времени» от размерности задачи.

Основной результат — наличие класса задач (NPC), не допускающих (пока) «быстрых» алгоритмов решения. Проблема $P=NP$ до сих пор открыта.

Этой теме будет посвящено две лекции в данной дисциплине.

3.Практический анализ алгоритмов

Результаты асимптотического анализа из теории сложности позволяют говорить о предпочтении в выборе алгоритмов при достаточно больших размерностях. Что делать в области реальных размерностей задач? Необходим переход к более детальному анализу алгоритмов, предполагающему введение модели вычислений (модели ЭВМ) и анализу в операциях этой модели.

Начало таким исследованиям положил Д. Э. Кнут в конце 1960-х в Искусстве программирования для ЭВМ. Его анализ основывался на введении абстрактной машины MIX и анализу алгоритмов в командах Ассемблера этой машины.

По сути, предлагаемая дисциплина и посвящена именно практическому анализу алгоритмов. Основной вопрос, на который мы хотим ответить — в каком диапазоне размерностей тот или иной алгоритм может быть рекомендован к применению в зависимости от его ресурсных характеристик.

4. Общая схема разработки алгоритмического обеспечения

1. Источником задачи является предметная область (по сути заказчики программного обеспечения). Мы получаем на вход процесса выбора и/или разработки алгоритмического обеспечения собственно задачу (например, сжатие изображений) и требования, которые, как правило имеют качественную формулировку.
2. С другой стороны, у нас есть множество алгоритмов решения предложенной задачи и нам необходимо решить задачу **рационального выбора алгоритма в заданных условиях его применения.**
3. Для такого выбора необходимо определить необходимые оценки качества алгоритмов (как универсальные, т.е. ресурсные оценки, так и проблемно ориентированные оценки — степень сжатия, качество восстановленного изображения, точность для итерационных методов и т.д.)

4. Для получения этих оценок необходимы методы анализ алгоритмов, и речь в дисциплине идет о знакомстве с различными методами анализа для получения ресурсных характеристик алгоритмов.

5. Важный практический момент — необходимо переформулировать требования заказчиков к программной системе в терминах оценок качества алгоритмов.

6. Получив оценки предполагаемых к использованию алгоритмов, мы можем решать задачу рационального выбора. Либо найдется такой алгоритм, который нас удовлетворяет, либо возникает вопрос о разработке нового алгоритма и/или модификации существующих.

Небольшой пример: Задача сжатия изображений вместе с требованиями большой степени сжатия при хорошем качестве без ограничений на время сжатия (задача создания электронной энциклопедии) приводит к выбору

алгоритма фрактального сжатия изображений. Большое время работы, сильное сжатие, быстрое восстановление.

А та же задача сжатия изображений *в видеопотоке* приведет очевидно к выбору совершенно другого алгоритма, поскольку здесь существенно ограничение на время — не более $1/24$ сек на кадр.

Если нет алгоритмов, удовлетворяющих заданным требованиям, то необходима разработка нового алгоритма (нужны методы разработки алгоритмов), или построение различных модификаций или комбинированных алгоритмов на основу уже существующих. При этом такие комбинированные алгоритмы должны опираться на результаты анализа. Нас интересуют такие свойства исходных данных и/или их характеристики (например размерность), при которых рационально применять именно данный алгоритм из множества рассматриваемых для решения поставленной задачи

5. Вопросы, рассматриваемые в дисциплине

- понятия алгоритма (по Э. Л. Посту) и модели вычислений;
- терминология и обозначения в теории ресурсной эффективности;
- ресурсные характеристики алгоритмов;
- методы анализа алгоритмов, в том числе и вероятностные методы;
- введение в теорию сложности вычислений;
- механизм обслуживания рекурсии и его ресурсные затраты;
- метод декомпозиции для разработки рекурсивных алгоритмов;
- основная теорема о рекуррентных соотношениях;
- примеры рекурсивных алгоритмов и их асимптотического анализа;
- методы построения комбинированных алгоритмов.