

**Атырауский инженерно-гуманитарный институт**

# **Алгоритмы**

**Подготовил: студент 2 курса специальности АиУ  
Отегенов Алтынбек**

**Проверила: старший преподаватель Кубашева Динара**

**Атырау-2018**

Понятие алгоритма относится к первоначальным, основным, базисным понятиям математики. Вычислительные процессы алгоритмического характера (арифметические действия над целыми числами, нахождение наибольшего общего делителя двух чисел и т. д.) известны человечеству с глубокой древности. Однако в явном виде понятие алгоритма сформировалось лишь в начале XX века.

Частичная формализация понятия алгоритма началась с попыток решения проблемы разрешения (нем. *Entscheidungsproblem*), которую сформулировал Давид Гильберт в 1928 году. Следующие этапы формализации были необходимы для определения эффективных вычислений или «эффективного метода»; среди таких формализаций — рекурсивные функции Геделя — Эрбрана — Клини 1930, 1934 и 1935 гг.,  $\lambda$ -исчисление Алонзо Чёрча 1936 г., «Формулировка 1» Эмиля Поста 1936 года и машина Тьюринга. В методологии алгоритм является базисным понятием и получает качественно новое понятие как оптимальности по мере приближения к прогнозируемому абсолюту. В современном мире алгоритм в формализованном выражении составляет основу образования на примерах, по подобию.

Современное формальное  
определение вычислительного  
алгоритма было дано в 30—50-е  
годы XX века в  
работах Тьюринга, Поста, Чёрча (тез  
ис Чёрча — Тьюринга), Н.  
Винера, А. А. Маркова.

## **Свойства алгоритмов**

Различные определения алгоритма в явной или неявной форме содержат следующий ряд общих требований:

Дискретность — алгоритм должен представлять процесс решения задачи как последовательное выполнение некоторых простых шагов. При этом для выполнения каждого шага алгоритма требуется конечный отрезок времени, то есть преобразование исходных данных в результат осуществляется во времени дискретно.

Детерминированность (определённость). В каждый момент времени следующий шаг работы однозначно определяется состоянием системы. Таким образом, алгоритм выдаёт один и тот же результат (ответ) для одних и тех же исходных данных. В современной трактовке у разных реализаций одного и того же алгоритма должен быть изоморфный граф. С другой стороны, существуют вероятностные алгоритмы, в которых следующий шаг работы зависит от текущего состояния системы и генерируемого случайного числа. Однако при включении метода генерации случайных чисел в список «исходных данных» вероятностный алгоритм становится подвидом обычного.

Понятность — алгоритм должен включать только те команды, которые доступны исполнителю и входят в его систему команд.

Завершаемость (конечность) – в более узком понимании алгоритма как математической функции, при правильно заданных начальных данных алгоритм должен завершать работу и выдавать результат за определённое число шагов. Дональд Кнут процедуру, которая удовлетворяет всем свойствам алгоритма, кроме, возможно, конечности, называет методом вычисления (англ. *computational method*). Однако довольно часто определение алгоритма не включает завершаемость за конечное время. В этом случае алгоритм (метод вычисления) определяет частичную функцию. Для вероятностных алгоритмов завершаемость как правило означает, что алгоритм выдаёт результат с вероятностью 1 для любых правильно заданных начальных данных (то есть может в некоторых случаях не завершиться, но вероятность этого должна быть равна 0).

Массовость (универсальность).

Алгоритм должен быть применим к разным наборам начальных данных.

Результативность — завершение алгоритма определёнными результатами.

## Виды алгоритмов

**Механические алгоритмы**, или

иначе **детерминированные, жесткие** (например, алгоритм работы машины, двигателя и т. п.) — задают определенные действия, обозначая их в единственной и достоверной последовательности, обеспечивая тем самым однозначный требуемый или искомый результат, если выполняются те условия процесса, задачи, для которых разработан алгоритм.

**Гибкие алгоритмы**, например, стохастические, то есть вероятностные и эвристические.

**Вероятностный** (стохастический) алгоритм дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.

**Эвристический алгоритм** (от греческого слова «эврика») — алгоритм, использующий различные разумные соображения без строгих обоснований.

**Линейный алгоритм** — набор команд (указаний), выполняемых последовательно во времени друг за другом.

**Разветвляющийся алгоритм** — алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько альтернативных ветвей алгоритма.

**Циклический алгоритм** — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. **Вспомогательный** (подчиненный) алгоритм (процедура) — алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи. **Структурная блок-схема, граф-схема алгоритма** — графическое изображение алгоритма в виде схемы связанных между собой с

# Представление алгоритмов

## Формы записи алгоритма:

словесная или вербальная (языковая, формульно-словесная);

псевдокод (формальные алгоритмические языки);

схематическая:

графическая (блок-схемы и ДРАКОН-схемы);

структурограммы (диаграммы Насси-Шнейдермана).

Обычно сначала (на уровне идеи) алгоритм описывается словами, но по мере приближения к реализации он обретает всё более формальные очертания и формулировку на

## Эффективность алгоритмов

Хотя в определении алгоритма требуется лишь конечность числа шагов, требуемых для достижения результата, на практике выполнение даже хотя бы миллиарда шагов является слишком медленным. Также обычно есть другие ограничения (на размер программы, на допустимые действия). В связи с этим вводят такие понятия, как сложность алгоритма (временная, по размеру программы, вычислительная и др.).

Для каждой задачи может существовать множество алгоритмов, приводящих к цели. Увеличение эффективности алгоритмов составляет одну из задач современной информатики. В 50-х гг. XX века появилась даже отдельная её область — быстрые алгоритмы. В частности, в известной всем с детства задаче об умножении десятичных чисел обнаружился ряд алгоритмов, позволяющих существенно (в асимптотическом смысле) ускорить нахождение произведения. См. быстрое умножение

Ярким примером является алгоритм Чудновского для вычисления числа  $\pi$ .