Лекция 1*. Алгоритм, определение, основные свойства. Необходимость уточнения понятия алгоритма.

п.1. Понятие алгоритма

Понятие алгоритма возникло задолго до появления ЭВМ и стало одним из основных понятий математики. Первое знакомство с алгоритмами происходит в начальной школе при изучении арифметически действий с натуральными числами. Понятие «алгоритм» давно является привычным не только для математиков. Оно составляет концептуальную основу разнообразных процессов обработки информации. Возможность автоматизации таких процессов обеспечивается наличием соответствующих алгоритмов. В упрощенном понимании «алгоритм» - это то, что можно запрограммировать на ЭВМ.

Слово алгоритм содержит в своем составе преобразованное географическое название Хорезм. Термин «алгоритм» обязан своим происхождением великому ученому средневекового Востока -Маххамад ибн Муса ал-Хорезми (Магомет, сын Моисея, из Хорезма). Он жил приблизительно с 783 по 850 гг., и в 1983 году отмечалось 1200-летие со дня его рождения в городе Ургенче областном центре современной Хорезмской области Узбекистана. В латинских переводах с арабского арифметического трактата ал-Хорезми его имя транскрибировалось как algorismi. Откуда и пошло слово «алгоритм» - сначала для обозначения алгоритмов цифровых вычислений десятичной позиционной арифметики, а затем для обозначения произвольных процессов, в которых искомые величины решаемых задач находятся последовательно из исходных данных по определенным правилам и инструкциям. Содержание понятия «алгоритм» можно определить следующим образом.

Алгоритм – предписание, однозначно задающее процесс преобразования исходной информации в виде последовательности элементарных дискретных шагов, приводящих за конечное число их применений к результату.

Алгоритмы возникли вместе с появлением математики. Школьный курс математики предлагает большой выбор алгоритмов:

- алгоритмы сложения и умножения "столбиком", деления "углом",
- приведения дробей к общему знаменателю,
- построения биссектрисы угла и т. д.

В высшей математике алгоритмов еще больше, причем наряду с численными алгоритмами (решение дифференциальных уравнений, задач *математического программирования* и др.) появляются алгоритмы над нечисловыми объектами (логическими выражениями, последовательностями произвольных символов, *графами* и т. п.).

Алгоритм является фундаментальным понятием информатики. Можно выделить три крупных класса алгоритмов:

- вычислительные,
- информационные,
- управляющие.

Вычислительные алгоритмы, как правило, работают со сравнительно простыми видами данных (числа, матрицы), но сам процесс вычисления может быть долгим и сложным.

---

* *Примечание.* В лекциях использованы материалы книги НОСОВ В.А. Основы теории алгоритмов и анализа их сложности. Курс лекций. Москва. 1992 г.

Информационные алгоритмы представляют собой набор сравнительно простых процедур (например, поиск числа или слова, удовлетворяющего определенным признакам), но работающих с большими объемами информации. Таковы алгоритмы в различных базах данных. Для того чтобы они работали эффективно, важно иметь хорошую организацию данных. Например, чтобы в картотеке можно было быстро найти нужные сведения, эти сведения нужно постоянно поддерживать в определенном порядке (по разделам, внутри разделов по алфавиту и т. д.).

Управляющие алгоритмы характеризуются тем, что данные к ним поступают от внешних процессов, которыми они управляют. Результаты работы этих алгоритмов представляют собой различные управляющие воздействия. Поэтому значения данных в ходе работы управляющих алгоритмов меняются (иногда очень быстро), и алгоритм должен вовремя правильно отреагировать, т.е. выдать нужный управляющий сигнал в нужный момент.

п.2. Необходимость уточнения понятия алгоритма

Алгоритм — это точная инструкция, а инструкции встречаются практически во всех областях человеческой деятельности. Возможны алгоритмы проведения физического эксперимента, сборки шкафа или телевизора, обработки детали. Однако не всякая инструкция есть алгоритм. Инструкция становится алгоритмом только тогда, когда она удовлетворяет определенным требованиям (свойствам):

1. **массовость** - способность обеспечить решение любой задачи из класса однотипных задач;
2. **понятность** - знание исполнителя о том, что надо делать для исполнения данного алгоритма;
3. **дискретность** - алгоритм должен быть представлен в виде конечной последовательности шагов (выполнение очередного шага начинается после завершения предыдущего);
4. **конечность** - выполнение алгоритма заканчивается после выполнения конечного числа шагов;
5. **результативность** - алгоритм должен приводить к решению задачи (или сообщать об отсутствии решения) за **конечное** число шагов;
6. **детерминированность** (определенность) - независимость от того, кто и сколько раз будет исполнять алгоритм. То есть формулировка алгоритма должна быть так точна, чтобы полностью определять все действия алгоритма.

Эти требования частично сформулированы вначале, хотя упомянутые в определении понятия однозначности и элементарности сами нуждаются в уточнении. Алгоритм однозначен, если при применении к одним и тем же данным он дает один и тот же результат. Но как по описанию алгоритма определить, однозначен он или нет? В каком случае шаги считаются элементарными?

Может возникнуть встречный вопрос: а так ли уж необходимо иметь точное определение алгоритма? В конце концов, в течение более двух тысячелетий люди создавали различные алгоритмы, не задумываясь над тем, что такое алгоритм вообще. Ответ на этот вопрос связан с двумя аспектами: теоретическим и практическим.

Вплоть до 30-х годов 20-го столетия понятие алгоритма оставалось интуитивным понятием, имевшим скорее методологическое, а не математическое значение. Так, к началу 20-го века много ярких примеров алгоритмов дала алгебра и теория чисел. Среди них упомянем алгоритм Евклида нахождения наибольшего общего делителя двух натуральных чисел или двух целочисленных многочленов, алгоритм Гаусса решения системы линейных уравнений, алгоритм Штурма для определения числа действительных корней многочлена с действительными коэффициентами на некотором отрезке действительных чисел, алгоритм разложения многочлена одного переменного над конечным полем на неприводимые множители. Указанные алгоритмические проблемы решены путем указания конкретных разрешающих процедур. Для получения результатов такого типа достаточно интуитивного понятия алгоритма. Под алгоритмом понималась процедура, которая позволяла путем выполнения последовательности элементарных шагов получать однозначный результат (не зависящий от того, кто именно выполнял эти шаги) или за конечное число шагов прийти к выводу о том, что решения не существует.

Долгое время речь шла лишь об алгоритмах, производящих вычисления и набор элементарных шагов был ясен. В него входили все арифметические операции, а также проверки равенств, неравенств или других отношений подобного типа. Часто алгоритмы задавались в виде формул, в которых с помощью скобок определялся порядок выполнения элементарных шагов в процессе вычисления.

К началу 20-го в. стали усложняться объекты, с которыми оперировали алгоритмы. Появилась необходимость выполнять операции над векторами, матрицами, множествами, функциями и т.п.

Возник ряд вопросов. Одни из них были связаны с трактовкой элементарности тех или иных шагов (например, можно ли считать взятие интеграла таким шагом?), другие - с оценкой конечности и однозначности алгоритма. В начале 20-го века были сформулированы алгоритмические проблемы, положительное решение которых представлялось маловероятным. Решение таких проблем потребовало привлечения новых логических средств.

С точки зрения самой математики желательно иметь дело только с точно определенными понятиями. Это нужно, например, для того, чтобы утверждать о некотором процессе, что он является алгоритмом или не является таковым. Точное определение важно и тогда, когда возникает желание доказать что-либо, относящееся к свойствам алгоритмов или к их возможностям. Ведь одно дело доказать существование разрешающего алгоритма - это можно сделать, используя интуитивное понятие алгоритма. Другое дело доказать отсутствие алгоритма – для этого нужно знать точно, что такое алгоритм.

Так возникла необходимость в точном понятии "любой алгоритм", т.е. максимально общем понятии алгоритма, под которое подходили бы любые мыслимые виды алгоритмов.

Если говорить о практическом аспекте, связанном с понятием алгоритма и с широким использованием этого понятия в *программировании* и *вычислительной технике*, то всякий алгоритм есть некоторая программа действия, а машинные программы с этой точки зрения и есть точные инструкции на выполнение некоторых процедур. Это наводит на мысль, что программы — это записи некоторых алгоритмов, а программирование есть процесс перевода записей алгоритмов на язык, понятный машине.

От программ мы ожидаем практической результативности. Вряд ли могут представлять интерес программы, которые никогда не приводят к результату. Поэтому и в практическом аспекте важно уточнить понятие алгоритма.

Возвратимся к тому словесному определению, которое было приведено в начале. Если используемые в нем на интуитивном уровне понятия однозначности, элементарности и результативности попытаться определить через какие-то другие понятия, то они, в свою очередь, также потребуют уточнения. Получается замкнутый круг. Чтобы вырваться из него, можно использовать следующий путь. Исходно задается лишь общая схема определения алгоритма. А ее детализация производится с помощью конкретного набора средств, которыми разрешается пользоваться в рамках данной алгоритмической модели.

п.3. Вычислимая функция и разрешимое множество

Всякому алгоритму соответствует задача, для решения которой он предназначен. В обратную сторону соответствие неоднозначно: задача может решаться различными алгоритмами. Алгоритмы, решающие одну и ту же задачу, называются эквивалентными. В теории алгоритмов все задачи принято делить на два больших класса:

- задачи, связанные с вычислением функций,
- задачи, связанные с распознаванием принадлежности объекта заданному множеству (что равносильно получению ответа на вопрос: *обладает ли объект заданным свойством*?).

В первом случае алгоритм A вычисляет функцию f_A, т.е., начав работать с входными данными x, он должен через некоторое время остановиться и выдать результат $y = f_A(x)$. Функция f_A не обязательно числовая. Например, телефонный справочник задает функцию, аргументы которой – имена или названия абонентов, а значения – их телефоны. Алгоритм вычисления этой функции заключается в поиске нужного имени с использованием алфавитного указателя.

Во втором случае алгоритм отвечает на вопрос: "Истинно ли высказывание $x \in M$?", или, что то же самое, проверяет истинность предиката (см. *Математическая логика*) $x \in M$ и выдает один из двух возможных результатов: *да* или *нет*. Второй случай можно считать разновидностью первого, поскольку предикат – это функция, принимающая два значения (истина или ложь). К тому же многие практические задачи объединяют в себе оба случая.

Например, при решении квадратного уравнения сначала нужно выяснить вопрос о существовании действительных корней уравнения (т.е. вопрос о принадлежности данного квадратного уравнения множеству уравнений, имеющих такие корни), который сводится к вопросу о знаке дискриминанта, и только при положительном ответе (дискриминант неотрицателен) вычисляются сами корни.

Тем не менее, разделение этих двух классов задач полезно, так как оно приводит к двум важным понятиям теории алгоритмов – **«вычислимая функция»** и **«разрешимое множество»**.

Функция называется **вычислимой**, если существует алгоритм, ее вычисляющий. **Множество** называется **разрешимым**, если существует алгоритм, который для любого объекта определяет, принадлежит он этому множеству или нет.