

Алгоритмы

Изучив эту тему, вы узнаете:

- ✓ в чем состоит назначение алгоритма и каковы его основные свойства;*
- ✓ какие типовые конструкции алгоритма существуют;*
- ✓ как представить алгоритм в виде блок-схемы;*
- ✓ каковы стадии разработки алгоритма.*



Понятие алгоритма

Знакомство с понятием алгоритма начнем с рассмотрения примера. Предположим, вы хотите вылепить из пластилина дракона. Результат многим будет зависеть от вашего знания и опыта. Однако достичь данной цели окажется гораздо легче если вы предварительно наметите действия, например следующие:

1. Изучить образ дракона по имеющейся картинке.
2. Вылепить голову.
3. Вылепить туловище.
4. Вылепить хвост.
5. Вытащить четыре ноги.
6. Сравнивая с картинкой, уточнить детали каждой вылепленной части дракона.

Следуя подготовленному плану, любой человек, даже не обладающий художественными способностями, но имеющий терпение. Обязательно получит хороший результат. Подобный план с подробным описанием действий, необходимых для получения ожидаемого результата, получил название алгоритма.

Алгоритм - описание последовательности действий (план), строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.

Алгоритмизация - процесс разработки алгоритма (плана действий) для решения задачи.



Свойство алгоритма

Мир алгоритмов очень разнообразен. Несмотря на это, удастся выделить общие свойства, которыми обладает любой алгоритм. Обычно мы выполняем привычные действия не задумываясь, механически. Например, вы хорошо знаете, как открывать дверь ключом. Однако, чтобы научить этому малыша, придется четко разъяснить и сами действия, и порядок их выполнения:

1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ два раза против часовой стрелки.
4. Вынуть ключ.

Представьте себе, что вас пригласили в гости и подробно объяснили, как добраться:

1. Выйти из дома.
2. Повернуть направо.
3. Пройти два квартала до остановки.
4. Сесть в автобус №5, идущий к центру города.
5. Проехать три остановки.
6. Выйти из автобуса.
7. Найти по указанному адресу дом и квартиру.

Это тоже не что иное, как алгоритм. Внимательно анализируя эти примеры, можно найти в них много общего, несмотря на значительное различие в сути самих действий. Эти общие характеристики называют свойствами алгоритма.



Свойство алгоритма

Для любого алгоритма характерны следующие свойства: дискретность, детерминированность, конечность, массовость, результативность.

Дискретность - это свойство указывает, что любой алгоритм должен состоять из конкретных действий, следующих в определенном порядке.

Детерминированность - это свойство указывает, что любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае.

Конечность - это свойство определяет, что каждое действие в отдельности и алгоритм в целом должны иметь возможность завершения.

Массовость - это свойство показывает, что один и тот же алгоритм можно использовать с разными исходными данными.

Результативность - это свойство требует, чтобы в алгоритме не было ошибок.



Линейный алгоритм

Описания действий в алгоритме следуют последовательно друг за другом. Однако очередность выполнения этих действий может быть изменена, если в алгоритме предусмотрен анализ некоторого условия. Путем включения условий создаются алгоритмы с различной структурой, в которой всегда можно выделить несколько типовых конструкций: линейную, циклическую, разветвляющуюся, вспомогательную.

Знакомство с типовыми конструкциями начнем с линейного алгоритма. Предположим, требуется составить алгоритм вычисления результата выражения:

$$100 + 15 - 40 + 20$$

1. Сложить числа 100 и 15.
2. Из полученной суммы вычесть 40.
3. К результату прибавить 20.

В этом примере действия выполняются в том порядке, в котором записаны. Подобные алгоритмы получили название линейных, или последовательных.

Линейный (последовательный) алгоритм - описание действий, которые выполняются однократно в заданном порядке.

Линейными являются алгоритмы отпирания дверей, заваривания чая, приготовления одного бутерброда. Линейный алгоритм применяется при вычислении арифметического выражения, если в нем используются только действия сложения и вычитания.



Циклический алгоритм

Многие процессы в окружающем мире основаны на многократном повторении одной и той же последовательности действий. Каждый год наступают весна, лето, осень и зима. Жизнь растений в течение года проходит одни и те же циклы. Подсчитывая число полных поворотов минутной или часовой стрелки, человек измеряет время. Допустим, робот обучен красить забор. Он последовательно закрашивает доску за доской. Для работы составлен следующий алгоритм:

1. *Покрасить доску.*
2. *Переместиться к следующей доске.*
3. *Перейти к действию 1.*

Робот, закрасив одну доску, перейдет ко второй, затем к следующей и т. д. Робот не сможет закончить работу, так как алгоритм не предусматривает окончания работы. В приведенном примере необходимо добавить в алгоритм действие по анализу результата:

1. *Покрасить доску.*
2. *Если есть еще доска, переместиться к следующей; перейти к действию 1.*
3. *Если доски закончились, завершить работу.*

Циклический алгоритм - описание действий, которые должны повторяться указанно число раз или пока не выполнено заданное условие.



Разветвляющийся алгоритм

Вспомним сюжет из русской сказки. Царевич останавливается у развилки дороги и видит камень с надписью: «Направо пойдешь - коня потеряешь, налево пойдешь - сам пропадешь.... Подобная ситуация, заставляющая нас принимать решение в зависимости от некоторого условия, постоянно встречается в повседневной жизни. Если пошел дождь, то надо открыть зонт. Если билет в кино стоит не больше десяти рублей, то купить билет и занять свое место в зале, иначе (если стоимость билета больше 10 руб.) вернуться домой. Если значение X больше нуля, то Y , равное $5 + X$, также больше нуля. Эти предложения начинаются с предположения о том, произошло или нет некоторое событие: пошел ли дождь, болит ли горло, прозвенел ли будильник и т. д. Приведенные при меры отражают суть нашего мышления. Делая какие-то предположения, мы неизбежно приходим к определенным выводам. Существует специальный раздел математики –формальная логика, которая объясняет, как выстраивать цепочку рассуждений, чтобы прийти к правильному выводу. Логика учит правильно формулировать условие, под которым понимается предположение, начинающееся со слова «если» и заканчивающееся перед словом «то». Условие может принимать значение. истина», когда оно выполнено, или «ложь», когда оно не выполнено. От значения условия зависит наше дальнейшее поведение. Например, в предложении «Если пошел дождь, то надо открыть зонт» условие «пошел дождь» может быть и истинным, и ложным. Поэтому в конкретной ситуации предполагается либо выполнение действия «открыть зонт», либо его пропуск - если дождя нет, то зонтик открывать незачем.

Условие - выражение, находящееся между словом «если» и словом «то» и принимающее значение «истина» или «ложь».



Разветвляющийся алгоритм

Разветвляющийся алгоритм - алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Итак, для того чтобы сделать выбор, надо проанализировать условие. В общем случае схема разветвляющего алгоритма будет выглядеть так: «если условие, то..., иначе...>). Такое представление алгоритма получило название полной формы.

Вспомните кота из сказки А. С. Пушкина: «идет направо - песнь заводит, налево сказку говорит». В разветвляющемся алгоритме при невыполнении условия действия могут не предусматриваться. Тогда это будет неполная форма, в которой действия пропускаются: «если условие, то...>). Неполная форма разветвляющегося алгоритма напоминает поведение водителя, едущего по шоссе: если бензин на исходе, то водитель заезжает на ближайшую автозаправочную станцию.



Вспомогательный алгоритм

Допустим, вы хотите научиться жонглировать двумя или даже тремя мячами. Если внимательно приглядеться к действиям профессионального артиста и попытаться понять, как это ему удается делать, то оказывается - секрет в том, что надо научиться искусно выполнять несколько определенных движений, которым присвоим соответствующие названия:

Бросок левой - подбросить мяч левой рукой.

Бросок правой - подбросить мяч правой рукой. Захват левой - поймать мяч правой рукой.

Захват правой - поймать мяч правой рукой.

Выполняться каждое такое действие будет по собственному алгоритму. Научившись таким действиям, вы сможете применить свое умение и в другом деле, например показывая фокусы или участвуя в соревнованиях. Благодаря тому, что подобные алгоритмы могут в дальнейшем многократно использоваться в других алгоритмах, их стали называть вспомогательными.

Понятие вспомогательного алгоритма значительно упрощает процесс алгоритмизации задачи. Создавая алгоритм, вы описываете действие, результатом которого должно быть достижение поставленной цели. Этому алгоритму можно дать уникальное имя.

Вспомогательный алгоритм - алгоритм, который можно использовать в других алгоритмах, указав только его имя. Вспомогательному алгоритму должно быть присвоено имя.



Стадии создания алгоритма

Человек легко читает и печатный, и рукописный текст. Однако написать алгоритм этого процесса так, чтобы он стал понятен компьютеру, - чрезвычайно непростая задача. В настоящее время уже разработан алгоритм распознавания компьютером печатного текста. А вот создать алгоритм, в соответствии с которым компьютер мог бы прочесть (распознать) рукописный текст, пока не удастся. Компьютер воспринимает такой текст как картинку.

Существует определенный алгоритм поведения водителей автомашин на дороге. Этот алгоритм определяется правилами дорожного движения, которые должен знать каждый водитель. Кроме того, шоферы особыми знаками предупреждают друг друга об опасности на дороге или подают сигналы помощи. Те же самые люди в другой обстановке будут общаться между собой совершенно по иному, в соответствии с принятыми для этой обстановки правилами поведения. Рассмотренные примеры говорят о том, что прежде всего алгоритм должен быть понятен человеку, а если возникает необходимость объяснить этот алгоритм другому человеку или объекту, то следует учитывать их особенности, в том числе среду, язык общения и пр. Разрабатывая алгоритм, вы всегда будете проходить минимум две стадии:

первая стадия - алгоритм должен быть представлен в форме, понятной человеку, который его разрабатывает;

вторая стадия - алгоритм должен быть представлен в форме, понятной тому объекту (в том числе и человеку), который будет выполнять описанные в алгоритме действия. В том случае, если эти действия станут выполнять сам разработчик алгоритма, вторая стадия будет отсутствовать.

