**Лекция 2. Понятие алгоритма**

Алгоритм (латинизов. Algorithmi, от имени персидского математика IX в. Аль-Хорезми) - последовательность, система, набор систематизированных правил выполнения вычислительного процесса, обязательно приводит к решению определенного класса задач после конечного числа операций. При написании компьютерных программ алгоритм описывает логическую последовательность операций. Для визуального изображения алгоритмов часто используют блок-схемы.

Единого «верного» определение понятия «алгоритм» нет.

«Алгоритм - это конечный набор правил, который определяет последовательность операций для решения конкретной множества задач и обладает пятью важными чертами: конечность, определенность, ввода, вывода, эффективность». (Д. Э. Кнут)

«Алгоритм - это всякая система вычислений, выполняемых по строго определенным правилам, что после какого-либо числа шагов заведомо приводит к решению поставленной задачи». (А. Колмогоров)

«Алгоритм - это точное предписание, определяющее вычислительный процесс, идущий от варьируемых исходных данных к искомому результату». (А. Марков)

«Алгоритм - точное предписание о выполнении в определенном порядке некоторой системы операций, ведущих к решению всех задач данного типа». (Философский словарь / Под ред. М. М. Розенталя)

«Алгоритм - строго детерминирована последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью понятных исполнителю команд». (Николай Дмитриевич Угринович, учебник «Информатика и информ. Технологии»)

**Происхождение термина алгоритм**

Слово алгоритм происходит от имени персидского ученого, астронома и математика Аль-Хорезми. Примерно 825 до н. е. он написал трактат, в котором описал придуманную в Индии позиционную десятичную систему счисления.

В первой половине XII века книга попала в Европу в переводе на латинский язык под названием Algoritmi de numero Indorum. Считается, что первое слово в переводе соответствует неудачной латинизации имени Аль-Хорезми, а название перевода звучит как «Алгоритм об индийской математике». Из-за неверного толкования слова Algoritmi как существительного во множественном числе ним стали называть метод вычисления.

неформальное определение

Каждый алгоритм предполагает существование начальных (входящих) данных и в результате работы приводит к получению определенного результата. Работа каждого алгоритма происходит путем выполнения последовательности некоторых элементарных действий. Эти действия называют шагами, а процесс их выполнения называют алгоритмическим процессом. Таким образом отмечают свойство дискретности алгоритма.

Важным свойством алгоритмов является массовость, или возможность применения к различным входных данных. То есть, каждый алгоритм призван решать класс однотипных задач.

Необходимым условием, которое удовлетворяет алгоритм, является детерминированность, или определенность. Это означает, что выполнение команд алгоритма происходит единственным способом и приводит к одинаковому результату для одинаковых входных данных.

Входные данные алгоритма могут быть ограничены набором допустимых входных данных. Применение алгоритма к недопустимым входных данных может приводить к тому, что алгоритм никогда не остановится, или попадет в тупиковое состояние (зависание) из которого сможет продолжить выполнение.

**Формальное Определение**

Различные теоретические проблемы математики и ускорения развития физики и техники поставили на повестку дня точное определение понятия алгоритма.

Первые попытки уточнения понятия алгоритма и его исследования осуществляли в первой половине XX века Тьюринг, Эмиль Пост, Жак Эрбран, Курт Гедель, Андрей Марков, Чёрч. Было разработано несколько определений понятия алгоритма, но впоследствии выяснилось, что все они определяют одно и то же понятие (см. Тезис Черча).

**Машина Тьюринга** Основная идея, лежащая в основе машины Тьюринга, очень проста.

Машина Тьюринга - это абстрактная машина (автомат), работающий с лентой отдельных ячеек, в которых записано символы.Машина также имеет головку для записи и чтения символов с ячеек, которая может двигаться вдоль ленты. На каждом шагу машина считывает символ из ячейки, на которую указывает головка, и, на основе считанного символа и внутреннего состояния делает следующий шаг. При этом, машина может изменять свое состояние, записать другой символ в ячейку, или передвинуть головку на одну ячейку влево или вправо. На основе исследования этих машин был выдвинут тезис Тьюринга (основная

гипотеза алгоритмов):

Для нахождения значений функции, заданной в некотором алфавите, тогда и только тогда существует некоторый алгоритм, когда функция равным по Тьюринга, то есть, когда ее можно вычислить на подходящей машине Тьюринга.

Этот тезис является аксиомой, постулатом, и не может быть доказана математическими методами, поскольку алгоритм не является точным математическим понятием.

**Алгоритмически неразрешимые задачи**

Функцию f называют вычислимой (англ. Computable), если существует машина Тьюринга, которая вычисляет значение f для всех элементов множества определения функции.

Если такой машины не существует, функцию f называют не вычислимой. Функция будет считаться не вычислимой даже если существуют машина Тьюринга, способны вычислить значение для подмножества из всего множества входных данных.

Случай, когда результатом вычисления функции f является булево значение истина или ложь (или множество {0, 1}) называют задачей, которая может быть разрешимой или неразрешимой в зависимости от вычислимости функции f.

Важно точно указывать допустимую множество входных данных, поскольку задача может быть разрешимой для одного множества и неразрешимой для другой.

Одной из первых задач, для которой была доказана неразрешимость проблема остановки. Формулируется она следующим образом:

Имея описание программы для машины Тьюринга, определить, завершит работу программа по конечное время, будет ли бесконечно, получив любые входные данные.

**Способы записи алгоритмов:**

1. графический (блок-схемы)

2. языковой

- Обычная речь (например., Кулинарный рецепт)

- псевдокод

Основные элементы схем алгоритма

Наименование

обозначение

функция

терминатор

Элемент отражает вход в программу или выход из нее (наиболее частое применение - начало и конец программы). Внутри фигуры записывается соответствующее действие.

процесс

Выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположение). Внутри фигуры записывают непосредственно сами операции.

решение

Показывает решение или функцию переключающего типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления понимал, определенных внутри этого элемента. Вход в элемент обозначается линией, входить

обычно в верхнюю вершину элемента. Если выходов два или три то обычно каждый выход сказывается линией, выходящей из остальных вершины (боковых и нижней). Если выходов более трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отражающие эти пути.

обусловленный процесс

Символ отображает выполнение процесса, состоящая из одной или нескольких операций, определенный в другом месте программы (в подпрограмме, модуле). Внутри символа записывается название процесса и переданы в него данные.

данные

Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Этот символ не определяет носителя (для указания типа носителя данных используются специфические символы).

**Подготовка**

Символ отображает модификацию команды или группы команд с целью воздействия на некоторую следующую функцию (установка переключателя, модификация индексного регистра или инициализация программы) Соединитель Символ отображает выход в часть схемы и вход из другой части этой схемы. Используется для обрыва линии и продолжения ее в другом месте (например: разделение блок-схемы, не помещается на листе). Соответствующие соединительные символы должны иметь одно (при том уникальное) обозначения.

комментарий

Используется для более подробной информации о шагах, процесса или группы процессов. Описание помещается со стороны квадратной скобки и охватывается ей по всей высоте. Пунктирная линия идет к описываемому элементу, или группе элементов (при этом группа выделяется замкнутой пунктирной линией). Также символ комментария следует использовать в тих случаях, когда объем текста в любом другом символе (например, символ процесса, символ данных и др) превышает его объем.

Псевдокод - компактная (чаще всего неформальная) язык описания алгоритмов, использует ключевые слова императивных языков программирования, но опускает несущественные подробности и специфический синтаксис.

Псевдокод конечно опускает детали, несущественные для понимания алгоритма человеком. Такими несущественными деталями могут быть описание переменных, системно-зависимый код и подпрограммы. Главная цель использования псевдокода - обеспечить понимание алгоритма человеком, сделать описание более воспринимаемым, чем исходный код на языке программирования. Псевдокод широко используется в учебниках и научно-технических публикациях, а также на начальных стадиях разработки компьютерных программ.

Базовые управляющие структуры

Название структуры

**Псевдокод**

присваивания, ввода, вывода

переменная = 0, введение (сменная), вывода (сменная)

разветвления

если условие то (серия1 иначе серия 2)

цикл ПОКА

пока условие нач серия кон

Пример программы «Hello, world!»

алг HELLOWORLD начале вывода ( "Hello, World!") конец алг HELLOWORLD

Общий вид алгоритма:

алг название алгоритма (аргументы и результаты)

дано условия применимости алгоритма

надо цель выполнения алгоритма

нач описание промежуточных величин

| последовательность команд (тело алгоритма)

кон

алг Сумма квадратов (арг целый n, рез целый S)

дано | n> 0

надо | S = 1 \* 1 + 2 \* 2 + 3 \* 3 + ... + n \* n

нач целый i

введение n; S = 0

пч для i от 1 до n

S = S + i \* i

инъекций

вывод "S =" S

кон