**Теория алгоритмов.**

**Определение алгоритма**

Характерной чертой современности является широкое использование компьютеров при решении проблем (задач) в различных областях человеческой деятельности. Однако предварительно задача должна быть решена алгоритмически, т.е. должно быть задано формальное предписание, следуя которому можно получить итоговый результат для решения всех задач определенного типа (это интуитивное, не строгое понятие алгоритма). Например, алгоритм нахождения наибольшего общего делителя двух натуральных чисел **а, b**, выглядит следующим образом:

1) разложить число **a**на простые множители;

2) повторить п. 1 для **b***и* перейти к п. 3;

3) составить произведение общих простых множителей из разложений **а** и **b** с показателями, равными наименьшим из показателей вхождения в разложения.

Проанализировав этот пример, отметим важнейшие черты (свойства) алгоритма:

**1. Массовость** — применимость алгоритма не к одной задаче, а к классу задач.

**2. Дискретность** — четкая разбивка на отдельные этапы (шаги) алгоритма.

**3. Детерминированность**— такая организация этапов выполнения, при которой всегда ясно как осуществить переход от одного этапа к другому.

**4. Конечность** — для получения результата при применении алгоритма к решению конкретной задачи выполняется конечная последовательность шагов алгоритма:

Отметим, что если наличие алгоритма само по себе служит доказательством существования алгоритма, то для доказательства его отсутствия необходимо иметь строгое математическое определение алгоритма.

Попытки формализовать понятие алгоритма привели к созданию **машины Тьюринга**, как некоего воображаемого устройства, реализующего алгоритм. Ещё одним шагом в определении понятия алгоритма стало появление **рекурсивных функций*,*** как функций, формализующих понятие алгоритма и реализующих интуитивное понятие вычислимости. Вскоре было установлено, что множество рекурсивных функций совпадает с множеством функций, вычислимых на машинах Тьюринга. Появлявшиеся затем новые понятия, претендующие на объяснение понятия алгоритма, оказывались эквивалентными функциям, вычислимым на машинах Тьюринга, а значит, и рекурсивным функциям. Итогом развернувшейся дискуссии о том, что такое алгоритм, стало утверждение, называемое сейчас тезисом Чёрча.

**Тезис Чёрча.** Понятие алгоритма, или вычислимости некоторым механическим устройством, совпадает с понятием вычислимости на машинах Тьюринга (а значит, с понятием рекурсивной функции). Другими словами, алгоритм это процесс, который может быть представлен функциональной схемой и реализован некоторой машиной Тьюринга.

Это утверждение нельзя считать математической теоремой. Это есть некоторый естественнонаучный тезис, принятый большинством исследователей.

**§6.2. Машина Тьюринга**

Машина Тьюринга представляет собой (абстрактное) устройство, состоящее из ленты, управляющего устройства и считывающей головки.

Лента разбита на ячейки. Во всякой ячейке в точности один символ из **внешнего алфавита** **A={a0,a1,…,an}.** Некоторый символ (будем обозначать его ) алфавита A называется пустым, а любая ячейка, содержащая в данный момент пустой символ, называется пустой ячейкой (в этот момент). Лента предполагается потенциально неограниченной в обе стороны.

**Управляющее устройство** в каждый момент времени находится в некотором состоянии qj, принадлежащем множеству **Q={q0, q1,..., qm}** (m=l). Множество Q называется **внутренним алфавитом***.* Одно из таких состояний (обычно **q0**) называется заключительным, а некоторое другое (обычно **q1**) -начальным.

Считывающая головка перемещается вдоль ленты так, что в каждый момент времени она обозревает ровно одну ячейку ленты. Головка может считывать содержимое обозреваемой ячейки и записывать в нее вместо обозреваемого символа некоторый новый символиз внешнего алфавита **A** (возможно тот же самый).

В процессе работы управляющее устройство в зависимости от состояния, в котором оно находится и символа, обозреваемого головкой, изменяет свое внутреннее состояние **q**. Затем выдает головке приказ напечатать в обозреваемой ячейке определенный символ из внешнего алфавита **A,** а потом приказывает головке либо остаться на месте, либо сдвинуться на одну ячейку вправо, либо сдвинуться на одну ячейку влево. Попав в заключительное состояние, машина прекращает работу.

**Конфигурацией на ленте (или машинным словом)** называется совокупность, образованная:

1) последовательностью символов из внешнего алфавита **А**, записанных в ячейках ленты, где **ai(1)**.- символ записанный в первой ячейке слева и т.д. (любая такая последовательность называется **словом),**

2) состоянием внутренней памяти qr;

3) номером **k** воспринимаемой ячейки.

Конфигурацию машины будем записывать так:

здесь **r**-воспринимаемая ячейка, выделяется в виде дроби.

Если машина, находясь во внутреннем состоянии **qi**, воспринимает ячейку с символом **au**, записывает к следующему моменту времени в эту ячейку символ **ar** , переходит во внутреннее состояние **qs** и сдвигается по ленте, то говорят, что машина выполняет команду **qiauqsarS** , где символ S может принять одно из значений: -1 – сдвиг головки влево; +1 - сдвиг головки вправо; 0 – головка остается на месте. Список всех команд (пятерок), определяющих работу машины Тьюринга, называется **программой** этой машины. Программу машины часто задают в виде таблицы. Так для описанной выше ситуации в таблице на пересечении строки **ai**и столбца **qi** будет стоять **qsarS**( см. таб.6.2.1)

**Таблица 6.2.1.**

**q0**

…

**qi**

…

**qm**

…

**au**

**qsarS**

…

Если в программе машины для пары **(qi,au)** пятерка отсутствует, то в таблице на пересечении строки **au**, и столбца **qi** ставится прочерк.

Итак, **машина Тьюринга есть, по определению**, набор **М=(А,Q,П)**, где **А** ― внешний алфавит, **Q** ― алфавит внутренних состояний, **П** ― программа.

Если машина, начав работу с некоторым словом P, записанным на ленте, придет в заключительное состояние, то она называется **применимой к этому слову**. Результатом ее работы считается слово, записанное на ленте в заключительном состоянии. В противном случае, машина называется не применимой к слову Р.

**Пример 6.2.2.**Построить машину Тьюринга, удваивающую натуральные числа, записанные в унарной системе счисления.

**Решение.** Искомую машину построим в алфавите A={,, }. Программа такой машины может выглядеть так:



Применим полученную машину к слову.



Введение новой буквы и замена исходных **|** на позволяет различить исходные **|**и новые (приписанные) **|**. Состояние **q1**обеспечивает замену**|** на *,* состояние **q2** обеспечивает поиск *,* предназначенных для замены на **|**, и останов машины в случае, когда не обнаружено, **q3** обеспечивает дописывание **|** в случае, когда произошла замена на**|.**