

Национальный исследовательский ядерный  
университет  
«МИФИ»

# Дискретная математика: теория алгоритмов и сложность вычислений

к.т.н., доцент Тихомирова Анна  
Николаевна

3 семестр

разделы	недели	мероприятия	баллы
Раздел 1 Формальные описания алгоритмов	1-6	КР1, КР2, Т1	22
Раздел 2 Числовые множества и арифметические вычисления	7-11	Т2	12
Раздел 3 Рекурсивные функции и сложность вычислений	12-15	КР3, Т3	16
Устная часть (зачет \ экзамен)			10 \ 25
Автоматизированная часть (зачет \ экзамен)			20 \ 25
Письменная часть (экзамен)			20
<b>Итого за семестр</b>			<b>100</b>

недели	мероприятия	баллы
3(4)	ЛР №1 (КР1) – м. Тьюринга	7
5(6)	ЛР №2 (КР2) – а. Маркова	7
7(8)	Тест №1 (Т1) – теория 1 раздела	5
11 (12)	Тест №2 (Т2) – теория 2 раздела	9
13 (14)	ЛР №3 (КР3) – Рекурсии	7
15 (16)	Тест №3 (Т3) – теория 3 раздела	6

# СИСТЕМА ПОДДЕРЖКИ ПРОЦЕССА ОБУЧЕНИЯ

[www.mephi22.ru](http://www.mephi22.ru)



Библиотека

ЛЕКЦИИ ▾ КНИГИ ПО ТА ▾ ОПРЕДЕЛЕНИЯ И ТЕОРЕМЫ ▾ К ЭКЗАМЕНУ ▾ ЗАДАЧИ ▾ ПЕРСОНАЛИИ ДОПОЛНИТЕЛЬНО

1 раздел

ЛЕКЦИЯ 1 ▾

ЛЕКЦИЯ 2 ▾

ЛЕКЦИЯ 3 ▾

ЛЕКЦИЯ 4 ▾

ЛЕКЦИЯ 5 ▾

2 раздел

ЛЕКЦИЯ 6 ▾

ЛЕКЦИЯ 7 ▾

ЛЕКЦИЯ 8 ▾

ЛЕКЦИЯ 9 ▾

ЛЕКЦИЯ 10 ▾

3 раздел

ЛЕКЦИЯ 12 ▾

ЛЕКЦИЯ 13 ▾

ЛЕКЦИЯ 14 ▾

ЛЕКЦИЯ 15 ▾

4 раздел

ЛЕКЦИЯ 16 ▾

clip2net.com

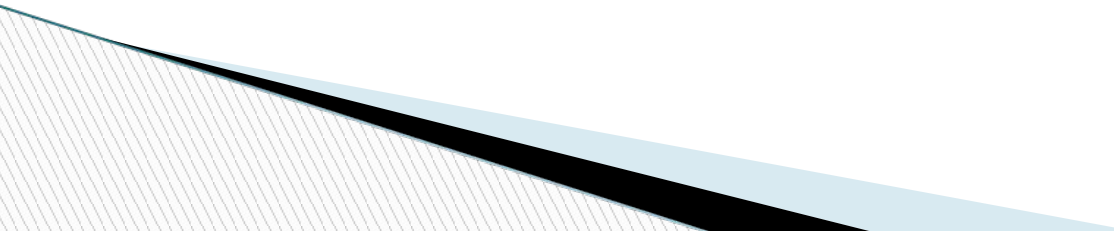
Всем необходимо зарегистрироваться до **11 сентября 2018 г**



# Лекция 1.

## Основные понятия

# *Причины появления теории*

- Внутренние потребности теоретической математики (математическая логика, алгебра, геометрия и анализ).
  - Создание быстродействующих электронных вычислительных и управляющих машин.
  - Связь с рядом областей лингвистики, экономики, физиологии мозга и психологии, философии, естествознания.
- 

# *Задача нахождения единообразной формы записи алгоритмов*

- Определение алгоритма через понятие вычислительной машины (машины Тьюринга, предложено Тьюрингом в 1937г. и машины Поста в то же время);
- Определение алгоритма через процедуру переработки слов по заданным правилам (нормальные алгоритмы, предложены Марковым в 1956г.);
- Определение алгоритма через рекурсивную функцию (предложено Клини и Гедделем в 1936г.).

# Алгоритмы: определение и основные свойства



слово “**алгоритм**” является производным от имени среднеазиатского ученого Аль Хорезми, уроженца Хивы, жившего в IX веке нашей эры.

**Алгоритм** – это точное предписание о выполнении в некотором порядке системы операций, определяющих процесс перехода от исходных данных к искомому результату для решения задачи данного типа.

Это понятие относится к исходным математическим понятиям, которые не могут быть определены через другие, более простые понятия.



# Свойства и параметры

Предписание считается алгоритмом, если оно обладает следующими свойствами:

- ▣ *определенностью*, то есть общепонятностью и точностью, не оставляющими место произволу,
- ▣ *массовостью*, то есть возможностью исходить из меняющихся в известных пределах значений исходных данных,
- ▣ *результативностью*, то есть направленностью на получение искомого результата,
- ▣ *элементарностью шагов*, на которые разбивается последовательность операций.

Каждый алгоритм, в общем случае, должен задаваться следующими параметрами:

- ▣ множеством допустимых исходных данных,
- ▣ начальным состоянием,
- ▣ множеством допустимых промежуточных состояний,
- ▣ правилами перехода из одного состояния в другое,
- ▣ множеством конечных результатов,
- ▣ конечным состоянием.

# Всеобщий алгоритм

"А нельзя ли создать **Всеобщий Алгоритм**, который решал бы любую математическую задачу аксиоматической теории?"

Готфрид Вильгельм **Лейбниц**:  
такой алгоритм будет найден!

## 1920е годы: две точки

Все проблемы алгоритмически разрешимы, но для некоторых алгоритм еще не найден, поскольку еще не развиты соответствующие разделы математики.

**зрения**

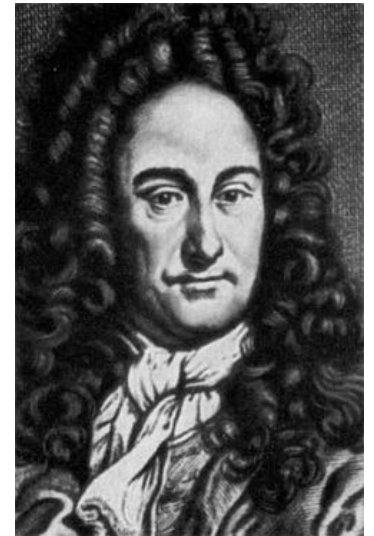
Есть проблемы, для которых алгоритм вообще не может существовать.

задача точного определения  
понятия алгоритма

# Историческая справка

Готфрид Вильгельм **Лейбниц**  
1646 —1716

немецкий философ, математик, физик



- создал математический анализ - дифференциальное и интегральное исчисления;
- создал комбинаторику как науку;
- заложил основы математической логики;
- описал двоичную систему счисления с цифрами 0 и 1,
- в механике ввёл понятие «живой силы» (прообраз современного понятия кинетической энергии) и сформулировал закон сохранения энергии.

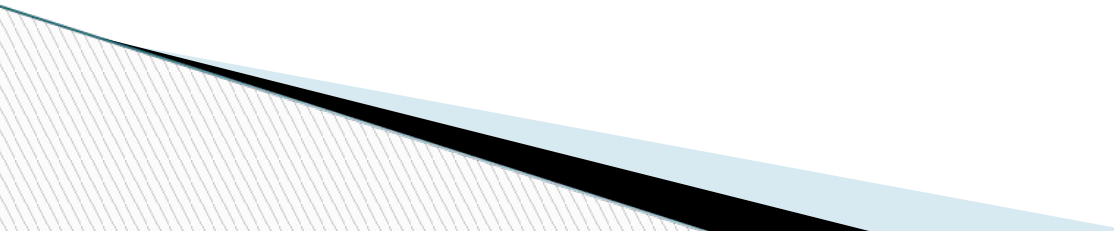
# Неразрешимые проблемы - пример

*Найти алгоритм, определяющий для любого диафантова уравнения, имеет ли оно целочисленное решение.*

Диафантово уравнение есть уравнение вида  $F(x, y, \dots, z) = 0$ , где  $F(x, y, \dots, z)$  — многочлен с целыми показателями степеней и с целыми коэффициентами.

В общем случае эта проблема долго оставалась нерешенной (в 1970 г. советский математик Ю. В. Матиясевич доказал ее неразрешимость).

# *Принципы построения машины*

- ❑ Каждый шаг алгоритма таков, что его может выполнить достаточно простое устройство (машина).
  - ❑ Желательно, чтобы это устройство было универсальным, т.е. чтобы на нем можно было выполнять любой алгоритм.
  - ❑ Механизм работы машины должен быть максимально простым по логической структуре, но настолько точным, чтобы эта структура могла служить предметом математического исследования.
- 

# Историческая справка

**Эмиль Леон Пост** (*Emil Leon Post*)  
1897 - 1954  
американский математик и логик



- один из основателей многозначной логики (1921);
- предложил абстрактную вычислительную машину - машину Поста.

# Машина Поста

Машина Поста — абстрактная вычислительная машина, состоящая из каретки (считывающей и записывающей головки) и ленты, разбитой на ячейки.

Каждая ячейка ленты может быть либо пустой («0»), или содержать метку («1»).

Программа состоит из пронумерованных строк. В каждой строке записывается одна из команд.

Каждая команда имеет следующий синтаксис:

$i. K j$

где  $i$  — номер команды,  $K$  — действие каретки,  $j$  — номер следующей команды (отсылка).

# Команды машины Поста

1.  $\rightarrow j$  – переместить каретку вправо на 1 ячейку и перейти к строке с номером  $j$
2.  $\leftarrow j$  – переместить каретку влево на 1 ячейку и перейти к строке с номером  $j$
3.  $1 j$  – записать в текущую ячейку «1» (поставить метку) и перейти к строке с номером  $j$
4.  $0 j$  – записать в текущую ячейку «0» (стереть метку) и перейти к строке с номером  $j$
5.  $? i; j$  – если текущая ячейка содержит «0» (не отмечена), то перейти к строке  $i$ , иначе перейти к строке  $j$
6.  $!$  – конец программы (стоп). В команде «стоп» переход на следующую строку не указывается



# Историческая справка

**Алан Тьюринг (Alan Mathison Turing)**

*1912 - 1954*

Английский математик, логик.



- Ввёл математическое понятие абстрактного эквивалента алгоритма, или вычислимой функции, получившее затем название «машины Тьюринга».
- Создал теорию «логических вычисляющих машин».
- Проводил исследования в области искусственного интеллекта.

# *Классические машины Тьюринга*

Задача описания алгоритма может быть сведена к построению машины некоторого типа, которая способна воспринимать набор правил, выраженных на некотором языке, и делать то, что предписано этими правилами.

А.М.Тьюринг, 1937 год

*Машина Тьюринга* – абстрактная (воображаемая) "вычислительная машина" некоторого точно охарактеризованного типа, дающая пригодное для целей математического рассмотрения уточнение общего интуитивного представления об алгоритме.

# Требования к машине Тьюринга



1. машина должна быть полностью детерминированной (вычисления должны быть точные и общепонятные) и действовать в соответствии с заданной системой правил.



2. должна допускать ввод различных “начальных данных” (соответствующих различным задачам из данного класса задач).



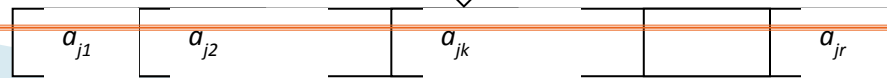
3. заданная система правил работы машины и класс решаемых задач должны быть согласованы так, чтобы всегда можно было “прочитать” результат работы машины.

**С помощью машины Тьюринга можно доказать существование или не существование алгоритмов решения различных задач.**

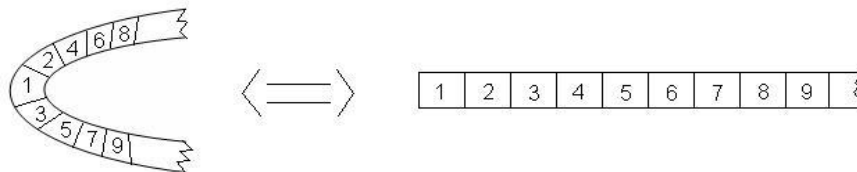
# Одноленточная машина Тьюринга

Под одноленточной машиной Тьюринга понимают кибернетическое устройство, состоящее из следующих элементов:

- бесконечной ленты, разделенной на ячейки,
- управляющей головки, способной читать символы, содержащиеся в ячейках ленты, и записывать символы в эти ячейки,
- выделенной ячейки памяти, содержащей символ внутреннего алфавита, задающий состояние машины Тьюринга,
- механического устройства управления, обеспечивающего перемещение головки относительно ленты,
- функциональной схемы — области памяти, содержащей команды (программу) машины Тьюринга (эта область доступна только для чтения).



# Лента



- Поскольку бесконечную ленту физически смоделировать затруднительно, обычно предполагается, что она конечная, и разбита на конечное число ячеек.
- В процессе работы к существующим ячейкам машина может пристраивать новые ячейки, так что лента может считаться потенциально неограниченной в обе стороны.
- Каждая ячейка ленты может находиться в одном из конечного множества состояний. Эти состояния мы будем обозначать символами  $a_0, a_1, \dots, a_m$  или другими символами. По сути это и есть символы, записанные в ячейках ленты. Совокупность таких символов называется внешним алфавитом машины, а сама лента часто называется внешней памятью машины.
- Если ячейка пустая, будем считать, что в ней записан условный символ  $\lambda$ .
- В процессе работы машины ячейки ленты могут изменять свое содержимое, но могут и не делать этого.
- Все вновь пристраиваемые ячейки пристраиваются пустыми (содержат условный символ  $\lambda$ ). Без ограничения общности ленту можно считать бесконечной лишь с одной стороны. В этом случае для удобства введем специальный символ  $\partial$ , обозначающий начало ленты. Этот символ имеет строго определенное место, его нельзя ни стирать, ни сдвигать. Тогда ленту можно считать однонаправленной (бесконечной вправо) и ее ячейки удобно просматривать слева направо.

# Управляющая головка

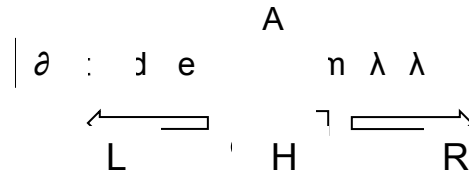
**Управляющая головка** – это некоторое устройство, которое может перемещаться вдоль ленты так, что в каждый рассматриваемый момент времени оно находится напротив определенной ячейки ленты.

Иногда, наоборот, считают управляющую головку неподвижной, а движущейся частью становится лента. В таком случае предполагается, что в управляющую головку вводится то одна, то другая ячейка ленты. Если какая-нибудь ячейка находится в управляющей головке, то говорят также, что машина в данный момент «воспринимает» или «обозревает» эту ячейку.

# Внутренняя память

**Внутренняя память машины** – это выделенная ячейка памяти, которая в каждый рассматриваемый момент находится в некотором «состоянии».

- Предполагается, что число возможных состояний внутренней памяти конечное и для каждой машины фиксированное.
- Состояние внутренней памяти мы будем обозначать символами  $S_0, S_1, \dots, S_n$ . Совокупность символов, обозначающих состояния внутренней памяти, называется **внутренним алфавитом** машины или **внутренними состояниями** машины.
- Одно из этих состояний называется **начальным**, с него начинает работу любая машина, пусть это будет состояние  $S_0$ . В процессе работы машина может какое-то количество шагов оставаться в состоянии  $S_0$  или возвращаться в него позднее.
- Еще одно специальное состояние – **заключительное**. Символ, обозначающий заключительное состояние, будет называться **стоп - символом**. Роль стоп - символа далее будет играть символ  $\Omega$ .



- Если в какой-то момент времени внутренняя память машины приходит в заключительное состояние  $\Omega$ , то дальнейших изменений в машине не происходит и машина называется **остановившейся**.
- Может случиться, что в машине никогда не будет происходить никаких изменений и при каком-то другом внутреннем состоянии  $S_i$ . Однако в этом случае мы будем говорить, что машина продолжает работать «вечно».
- В большинстве случаев е останавливающиеся машины не имеют никакой ценности, так как невозможно запротоколировать факт окончания выполнения алгоритма и, соответственно, считать полученный ответ. Обычно говорят, что, если машина  $T$  останавливается на входном слове  $t$ , то она к этому слову **неприменима**.

ВНЕШНИЙ АЛФАВИТ

- a, b, c, 0, 1, 2...
- $\partial$  - начальный символ
- $\lambda$  - пустой символ

ВНУТРЕННИЙ АЛФАВИТ

- A, B, C...
- $S_0$  - начальное состояние
- $\Omega$  - конечное состояние



# Механическое устройство

- Предполагается, что машина снабжена особым механизмом, который в зависимости от символа в воспринимаемой ячейке и состояния внутренней памяти может выполнить следующие действия:
  - изменить состояние внутренней памяти,
  - одновременно изменить содержимое воспринимаемой ячейки,
  - сдвинуть (вправо, влево) управляющую головку в соседнюю ячейку.
- В частном случае содержимое воспринимаемой ячейки и/или состояние внутренней памяти могут оставаться неизменными, а управляющая головка – неподвижной (H).
- Если управляющая головка находится в самой правой ячейке и по ходу работы машина должна сдвинуть управляющую головку в соседнюю справа (отсутствующую) ячейку, то предполагается, что, сдвигая головку, машина одновременно пристроит недостающую ячейку с пустым символом.
- Аналогично работает машина и в случае, когда головка воспринимает самую левую ячейку и по ходу дела машине надо сдвинуть головку еще левее. Впрочем, далее предполагается, что лента бесконечна вправо, а ее самая левая ячейка занята специальным символом начала ленты, обозначаемым  $\partial$ .

# Конфигурация м. Тьюринга

Конфигурация машины Тьюринга – совокупность, образованная содержимым текущей обозреваемой ячейки  $a_j$  и состоянием внутренней памяти  $S_i$ .

Работа машины состоит в том, что она из данного «состояния» по истечении одного такта работы механического устройства переходит в следующее «состояние», затем из этого состояния по истечении такта работы переходит в новое состояние и так далее.

Т.о. если машина, имея внутреннее состояние  $S_i$  и воспринимая ячейку ленты с символом  $a_j$ , изменяет свое внутреннее состояние на  $S_q$  и одновременно содержимое воспринимаемой ячейки заменяет символом  $a_y$ , а управляющая головка сдвинется на одну ячейку вправо (R), то говорят, что машина выполняет команду соответственно  $S_i a_j \rightarrow a_y R S_q$ .

Если управляющая головка сдвинется влево (L) или останется неподвижной (H), то команды записываются соответственно:

$$\begin{array}{l} S_i a_j \rightarrow a_y L S_q \\ S_i a_j \rightarrow a_y H S_q \end{array}$$

# Программа машины Тьюринга

Программа машины Тьюринга – совокупность команд установленного формата

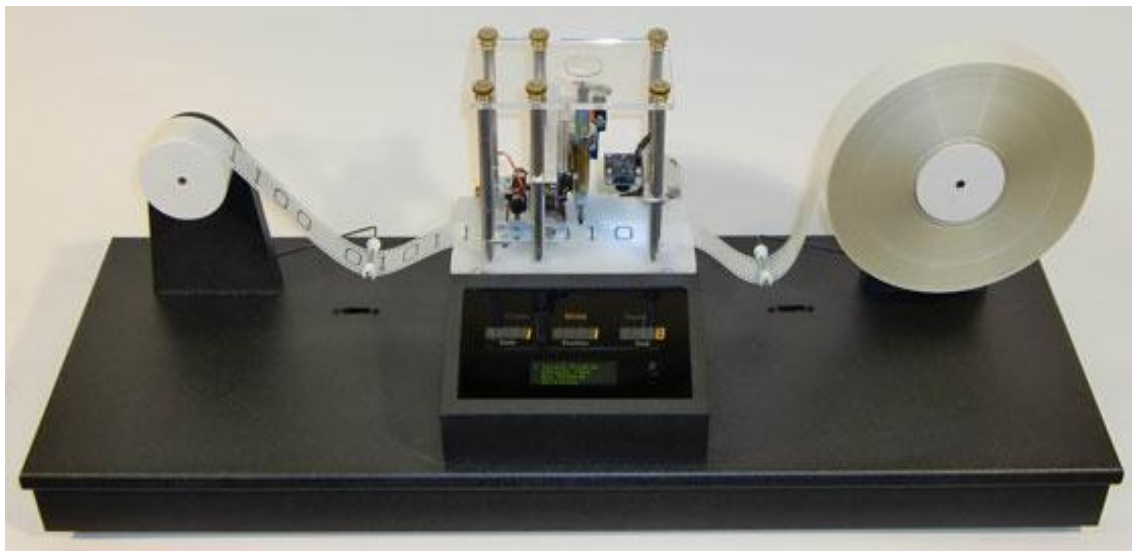
Так как работа машины по условию целиком определяется состоянием в данный момент ее внутренней памяти  $S_i$  и содержимым воспринимаемой ячейки  $a_j$ , то для каждой  $S_i a_j$  ( $i=1, \dots, n; j=1, \dots, m$ ), программа машины должна содержать одну и только одну команду, начинающуюся словом  $S_i a_j$ .

Программа машины с символами  $\{a_0, a_1, \dots, a_n\}$  и состояниями  $\{S_0, S_1, \dots, S_m\}$  содержит максимум  $n \cdot m$  команд.

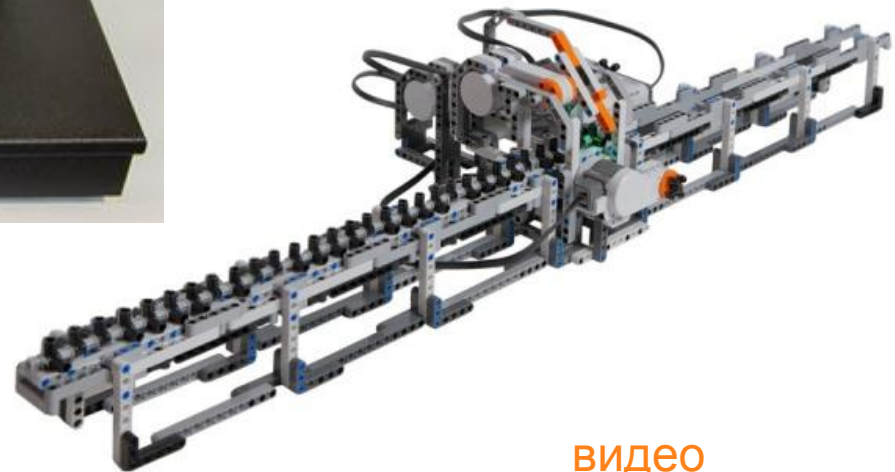
При этом некоторые команды являются «мертвыми», в том случае, если ни при каких входных словах в данном алфавите невозможно наступление этой конфигурации. В грамотной, с точки зрения реализации, программе таких строк быть не должно, хотя формально их наличие ошибкой не является.

# Реальные машины Тьюринга

Машина Тьюринга была построена в металле в 1973 в Малой Крымской Академии Наук.



<http://www.legoturingmachine.org/>



[ВИДЕО](#)

<http://aturingmachine.com/index.php>

# Тезис Тьюринга

Тезис Тьюринга – любой алгоритм можно преобразовать в машину Тьюринга.

Эту гипотезу невозможно доказать, потому что она оперирует неформальным понятием алгоритма.

Однако обоснование гипотезы есть: все алгоритмы, придуманные в течение столетий, могут быть реализованы на машине Тьюринга.

Кроме того, эквивалентность всех формальных определений алгоритма неслучайна и говорит в пользу гипотезы.

Чтобы опровергнуть основную гипотезу необходимо придумать такой алгоритм, который невозможно было бы реализовать на машине Тьюринга, но пока такого алгоритма нет.

# Литература

- Мальцев А.И. Алгоритмы и рекурсивные функции. М.: Наука, 1965, 1986.
- М.Минский. Вычисления и автоматы. М.: Мир, 1971
- Ахо А. Построение и анализ вычислительных алгоритмов. Пер. с англ. М.: Мир, 1979.
- Ахо А. Хопкрофт Д. Структуры данных и алгоритмы. Пер. с англ. М.: Издательский дом «Вильямс», 2000.
- Яблонский С.В. Введение в дискретную математику. М.: Наука, 1979; 1996.
- ПападимитриуХ., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. Пер. с англ. М.: Мир, 1985.
- Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов. М.: Наука, 1975, 1984.
- Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
- Гашков С.Б., Чубариков В.Н. Аримфетика. Алгоритмы. Сложность вычислений. М.: Высшая школа, 2000.
- Гуц А.К. Кардинальные и трансфинитные числа: Учебное пособие. Омск: Омск.университет, 1995.
- Хаусдорф Ф. Теория множеств. Пер. с нем. М.: КомКнига, 2006.
- Бурова И.Н. Парадоксы теории множеств и диалектика. М.: Наука, 1976.
- Виленкин Н.Я. Рассказы о множествах. М.: Наука, 1969.
- Коэн П.Дж. Теория множеств и континуум-гипотеза. Пер. с англ. М.: Мир, 1969.
- Успенский В.А., А.Л. Семенов. Теория алгоритмов: основные открытия и приложения. М.: Наука, 1987.
- Бурбаки Н. Теория множеств. Пер. с франц. М.: Мир, 1965.
- Горбатов В.А. Фундаментальные основы дискретной математики. М.: Наука, 2000.
- Горбатов В.А., Горбатов А.В., Горбатова М.В. Дискретная математика: учебник для студентов ВТУЗов. М.: АСТ: Астрель, 2006.
- Кантор Г. Труды по теории множеств. М.: Наука, 1985.
- Эщби У.Р. Введение в кибернетику. Пер. с англ. М.: КомКнига, 2005.
- Фалевич Б.Я. Теория алгоритмов: Учебное пособие. М.: Машиностроение, 2004.
- Пенроуз Р. Новый ум короля: о компьютерах, мышлении и законах физики. Пер. с англ. М.: Едиториал УРСС, 2005.
- Пойя Д. Математика и правдоподобные рассуждения. Пер. с англ. М.: Издательство иностранной литературы, 1957.
- Френкель А.А., Бар-Хиллел И. Основания теории множеств. Пер. с англ. М.: КомКнига, 2006.
- Гельфонд А.О., Трансцендентные и алгебраические числа. М.: КомКнига, 2006.
- Марков А.А., Нагорный Н.М. Теория алгоритмов. М.: ФАЗИС, 1996.