Профессиональное образование
в современном мире.
2019. Т. 9, №2, С. 2753–2764
DOI:10.15372/PEMW20190212
ISSN 2224–1841 (печатный)

Professional education in the modern world,
2019, vol. 9, no. 2, pp. 2753–2764
DOI: 10.15372/PEMW20190212
ISSN 2224–1841 (print)

О ЗНАЧЕНИИ ТЕОРИИ АЛГОРИТМОВ ДЛЯ СИСТЕМЫ СОВРЕМЕННОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И МЕТОДИКИ ЕЕ ПРЕПОДАВАНИЯ

THE SIGNIFICANCE OF THE THEORY OF ALGORITHMS FOR MODERN PROFESSIONAL EDUCATION AND METODOLOGY OF ITS TEACHING

УДК 51(072)

В.И. Игошин
Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского, Саратов, Россия,
e-mail: igoshinvi@mail.ru
ORCID https://orcid.org/0000-0003-0909-0009

Аннотация. *Статья посвящена выявлению роли и значения теории алгоритмов в фундаментализации математического образования специалистов в области компьютерных наук и информационных технологий, обучающихся в образовательных учреждениях СПО и ВО. При этом теория алгоритмов предстает в двух своих ипостасях: как теория конкретных алгоритмов (или интуитивно-содержательная теория алгоритмов) и как формально-логическая (абстрактная) теория алгоритмов. В первом случае теория алгоритмов занимается созданием и изучением алгоритмов решения конкретных задач, и главной проблемой здесь является проблема разработки такого конкретного алгоритма, который может быть реализован современным компьютером в реальное время, а также проблема сравнения различных конкретных алгоритмов решения одной и той же задачи по степени их сложности, в основном по времени, требуемом для решения задачи. В связи с этим возникают классы сложности алгоритмов P и NP, а вместе с ними и проблема взаимоотношений между этими классами, не решенная до конца до настоящего времени. Во втором случае теория алгоритмов создает строго математические (абстрактные) понятия алгоритма и изучает свойства таких понятий. В 1930-е годы и первые послевоенные годы было разработано несколько абстрактных понятий алгоритма или, как говорят, формализаций интуитивного понимания алгоритма. Это машины Тьюринга и вычислимые с их помощью функции, рекурсивные функции как функции вычислимые с помощью некоторого*

DOI: 10.15372/PEMW20190212

Igoshin, V. I.
Saratov State University,,
Saratov, the Russian Federation
e-mail: igoshinvi@mail.ru
ORCID https://orcid.org/0000-0003-0909-0009

Abstract. *The article explores the role and relevance of the theory of algorithms in the fundamentalization of mathematical education of specialists in the field of computer science and information technology, trained at vocational schools and higher institutions. In this case, the theory of algorithms appears in two fields; the first field assumes theory of specific algorithms (or intuitively-containing theory of algorithms) whereas the second one implies a formal-logical (abstract) theory of algorithms. In the first case, the theory of algorithms explores algorithms as means for solving specific problems, and the problem of developing this kind of algorithm is seen as the basic one. This algorithm should be implemented by a modern computer in real time. The second problem assumes comparing different specific algorithms for solving the same problem in terms of their complexity, mainly in terms of the time required to solve the problem. Due to this fact, the classes of complexity of algorithms P and NP appear followed by the problem of relationships between these classes. These problems are not solved until now. In the second case, the theory of algorithms creates mathematical (abstract) concepts of the algorithm and explores the properties of these concepts. In the 30s and the first postwar years of the XX century, several abstract concepts of the algorithm, or formalizations of the intuitive understanding of the algorithm, were developed. This is a Turing machine, recursive functions as functions computable by some algorithm, normal algorithms of A.A. Markov. The abstract theory of algorithms establishes the equivalence of these abstract con-*

алгоритма, нормальные алгоритмы А. А. Маркова и вычислимые с их помощью функции. Абстрактная теория алгоритмов устанавливает эквивалентность этих абстрактных понятий. Важнейшей проблемой здесь является также проблема существования таких алгоритмов для решения той или иной массовой проблемы. В частности, абстрактная теория алгоритмов устанавливает отсутствие алгоритмов для решения ряда массовых проблем. В нашей работе характеризуется методическая система обучения теории алгоритмов, учитывающая эти две ее ипостаси: интуитивно-содержательную и абстрактную.

Ключевые слова: *интуитивное понимание алгоритма, формальные теории алгоритмов, машины Тьюринга, рекурсивные функции, нормальные алгоритмы Маркова, компьютерные науки, информационные технологии, методика обучения теории алгоритмов.*

Для цитаты: *Игошин В. И.* О значении теории алгоритмов для системы современного профессионального образования и методике её преподавания // Профессиональное образование в современном мире. 2019. Т. 9, № 2. С. 2753–2764

DOI: 10.15372/PEMW20190212

cepts. The problem of these algorithms existing is seen as the most important one. In particular, the abstract theory of algorithms establishes the absence of algorithms for solving a number of mass problems. This paper describes a methodological system of teaching algorithm theory, taking into account its two intuitive and abstract fields.

Keywords: *Intuitive understanding of the algorithm, formal theories of algorithms, Turing machine, recursive functions, Markov normal algorithms, computer science, information technology, methods of teaching the theory of algorithms.*

For quote: *Igoshin V. I.* [The significance of the theory of algorithms for modern professional education and metodology of its teaching]. *Professionalnoe obrazovanie v sovremennom mire = Professional education in the modern world*, 2019, vol. 9, no. 2, pp. 2753–2764

DOI: 10.15372/PEMW20190212

Введение. Во второй половине XX века – начале XXI века на передний план среди прикладных разделов математической науки выдвигается так называемая дискретная математика. Это комплекс математических дисциплин, которые строят модели тех процессов и явлений окружающего мира, которые обладают свойством дискретности – противоположным свойству непрерывности. Важнейшей субстанцией, подвергаемой анализу с помощью методов дискретной математики, является информация. Именно разработка во второй половине XX века методов изучения информации привела к созданию фундаментальных разделов дискретной математики, которые способствовали формированию идеологии компьютеров, а затем и созданию самих компьютеров. Важнейшим таким разделом дискретной математики является теория алгоритмов. Именно эта научная дисциплина образует базу многочисленных языков программирования, которые являются основой «компьютерных мозгов». Напомним, что основой «компьютерного железа» являются методы, разработанные математической логикой. К компьютерным мозгам математическая логика имеет также самое непосредственное отношение.

Интуитивное понимание понятия алгоритма складывалось в практике, в науке и, прежде всего, в математике с древнейших времен. Алгоритм в таком понимании есть некая четкая система инструкций, которая будучи последовательно примененная к начальному набору каких-либо конструктивных объектов в результате за конечное число шагов приводит к созданию некоего результирующего конструктивного объекта. При таком интуитивном понимании понятия алгоритма можно говорить об алгоритмах в самых разнообразных областях человеческой деятельности.

Во второй половине XX века математика сделала понятие алгоритма объектом своего строгого изучения, охватив едиными абстрактными математическими понятиями все мыслимое многообразие конкретных алгоритмов, известных из практической деятельности. Это позволило теории алгоритмов наряду с развившейся к этому времени математической логикой стать фундаментальным основанием теории и практики создания и функционирования компьютеров. Именно поэтому в статье [1] отмечено, что важнейшими фундаментальными компонентами профессиональных компетенций в сфере компьютерных наук и информационных технологий являются знания и умения в области математической логики и теории алгоритмов, и выявлена роль курса математической логики в фундаментализации математического образования специалистов в области компьютерных наук и информационных технологий, обучающихся в образовательных учреждениях СПО, а также ВО.

На XIII Всемирном конгрессе по математическому образованию, проходившем в июле 2016 года в Гамбурге, отмечалось, что дискретная математика – это математика нашего времени и, в частности, обсуждался вопрос о преподавании разделов дискретной математики на различных образовательных уровнях [2; 3].

Изложенные обстоятельства обусловливают необходимость изучения основ теории алгоритмов в системе учреждений профессионального образования всеми, чья будущая профессия будет связана с компьютерами, информатикой, искусственным интеллектом. Эта необходимость, в свою очередь, порождает потребность в разработке такой методической системы обучения теории алгоритмов, которая позволила бы вооружить обучающихся основополагающими знаниями по данной дисциплине и ее приложениям к теории и практике программирования.

Постановка задачи. Целями исследования в работе являются, во-первых, выявление роли и значения теории алгоритмов как одного из разделов дискретной математики в фундаментализации математического образования специалистов в области компьютерных наук и информационных технологий, обучающихся в образовательных учреждениях СПО и ВО, и во-вторых, разработка основных положений методической системы обучения теории алгоритмов в этих учреждениях.

Методология и методика исследования. В работе применялись следующие методы исследования: изучение и анализ научной литературы по теории алгоритмов и ее применению к практике программирования и информатике; системный подход и сравнительный анализ, позволяющие рассматривать учебную дисциплину «Теория алгоритмов» в комплексе дискретно-математических дисциплин, а также в системе ее взаимосвязей со всем математическим и компьютерным образованием будущих специалистов в области компьютеров, информатики и искусственного интеллекта. Использовался практический опыт преподавания автором на протяжении трех десятков лет учебных дисциплин «Математическая логика» и «Теория алгоритмов».

Результаты. *Алгоритмы в технике и в науке.* Роль и значение алгоритмов в технике необычайно велики. По сути, вся технико-производственная деятельность людей носит алгоритмический характер. Алгоритмы такой деятельности содержатся в многочисленных справочниках для конструкторов, инженеров и техников, мастеров и квалифицированных рабочих, врачей, фельдшеров и медицинских сестер, архитекторов и строителей, бухгалтеров и т.д. Уже давно учёные и инженеры заметили, что если удалось получить алгоритм решения какой-нибудь задачи, то можно создать машину, которая решала бы эту задачу, т.е. можно автоматизировать её решение. Но алгоритм решения той или иной задачи не тождествен алгоритму управления машиной, решающей данную задачу. Эти алгоритмы в определённом смысле эквивалентны. Алгоритм решения задачи может отыскивать математик. Осуществлять же алгоритмизацию процесса управления машиной должен тот специалист, который компетентен в той области, откуда взята задача.

Содержательные явления, которые легли в основу образования понятия «алгоритм», издавна занимали важное место в науке. Алгоритмы встречаются в науке на каждом шагу, и их роль для науки чрезвычайно велика. Наука в своих рассмотрениях всегда стремится к максимальной общности, стремится решать задачи «в самом общем виде». В конечном виде такое умение почти всегда обращается в умение владения тем или иным алгоритмом. Говоря, например, об умении человека складывать числа, имеют в виду не то, что он для любых двух чисел рано или поздно сумеет найти их сумму, а то, что он владеет некоторым единообразным приёмом сложения, применимым к любым двум конкретным числам, т.е. владеет *алгоритмом* сложения двух чисел.

Понятие задачи «в общем виде» получает своё уточнение при помощи понятия «массовая проблема» или «массовая алгоритмическая проблема». Такая проблема задаётся бесконечной серией отдельных однотипных единичных задач и состоит в требовании найти единый алгоритм их решения. Многочисленные справочники по разнообразным научным дисциплинам в значительной мере заполнены алгоритмами решения разнообразных массовых проблем, возникающих в соответствующей области науки. Алгоритмы образуют своего рода «золотой запас» каждой научной дисциплины. Их значение для науки можно охарактеризовать следующими положениями: 1) алгоритмы являются формой изложения научных результатов; 2) они являются руководством к действию при решении уже изученных проблем и как следствие: 3) средством, позволяющим экономить умственные усилия и умственный труд; 4) они служат необходимым этапом при автоматизации решения задач; 5) алгоритмы являются средством (инструментом), используемым при исследовании и решении новых проблем; 6) алгоритмы предоставляют язык для описания разнообразных сложных процессов.

Здесь следует отметить, что, хотя алгоритмы составляют важную часть каждой науки, они конечно же не исчерпывают целиком её содержания. Не менее важны в науке понятия и их определения, входящие в данную науку, установленные ею факты (в математике – это доказанные теоремы), выработанный наукой подход к изучаемым объектам и явлениям.

Алгоритмы в образовании. Раз роль и значение алгоритмов столь велики в науке, технике и практике, то и сфера образования не может обойтись без алгоритмов. При этом в образовании алгоритмы играют двоякую роль. С одной стороны, в образовании на различных его уровнях и в различных его областях изучаются разнообразные конкретные алгоритмы. С другой стороны, сам образовательный процесс носит характер алгоритмического процесса. Рассмотрим эти две алгоритмические ипостаси образования.

Обучение алгоритмам, выработанным в той или иной области науки, является неотъемлемым атрибутом современного образования. Применение разработанных и хорошо известных алгоритмов решения тех или иных задач не требует особой творческой инициативы, но способствует выработке прочных навыков, необходимых в любой деятельности. Так, «музыкант должен тысячи раз повторять скучные пассажи. При изучении иностранного языка надо заучивать и много раз повторять новые слова. При изучении математики надо твердо заучить таблицу умножения, бегло производить выкладки…, бегло выполнять тригонометрические преобразования…» [4, с. 4]. Но при этом не только вырабатываются прочные навыки в той или иной сфере деятельности. В конечном итоге такое обучение приводит к формированию алгоритмических приёмов мышления или даже алгоритмического стиля мышления. Владение такими приёмами позволит человеку в будущем отыскивать единый общий метод решения целой серии однотипных задач, отличающихся друг от друга лишь значениями некоторых параметров, характеризующими начальные условия и ход решения задачи, что является чрезвычайно важным для соответствующих профессиональных компетенций. Умение применять алгоритмические приёмы мышления и поведения – непременное требование к инженерно-техническому и управленческому персоналу современных предприятий, фирм, офисов и других организаций.

На первый взгляд может показаться, что обучение алгоритмическим приёмам мышления и действия не развивает и даже подавляет творческие способности людей. Ведь исполнитель алгоритма действует механически, от него не требуется никакой фантазии, изобретательности или сообразительности; алгоритм просто не оставляет ему места для проявления этих качеств, если он ими обладает. На этапе исполнения алгоритмов это действительно так. Но ведь алгоритмы необходимо создавать, и эта задача обладает неисчерпаемым творческим потенциалом. Здесь – неограниченное поле для изобретательности, поиска, полёта фантазии. В особенности, когда требуется отыскать наилучший по каким-либо качествам алгоритм. Не научившись применять алгоритмы, не научившись владеть алгоритмическими приёмами, не овладев алгоритмическим стилем мышления, невозможно будет и создавать новые и улучшать имеющиеся алгоритмы, проявить на этом поприще свои творческие способности. Ещё раз подчеркнём, что после того, как алгоритм создан, проявление творческих фантазий и изобретений было бы неоправданным расходом психической энергии. Таким образом, в ходе обучения алгоритмы позволяют эту энергию учащихся экономить. В то же время, как мы отмечали выше, ни одна наука не исчерпывается только выработанными в ней алгоритмами, поэтому и обучение, конечно же, не может сводиться к обучению только алгоритмам и как следствие – к развитию только алгоритмического мышления в его чисто механическом понимании.

Вторая сторона использования алгоритмов в образовании – алгоритмизация самого процесса обучения. По существу, это один из путей применения точных математических методов в педагогической науке и практике. Интерес к такому направлению возник в связи с появлением в педагогике идеи подхода к обучению как к процессу управления, естественно возникшим под влиянием кибернетики. В 1960-е годы это направление нашло проявление в концепции программированного (программно-управляемого) обучения. Кибернетические идеи в педагогике и программированное обучение привели к введению в оборот теоретической и практической педагогики математических средств из арсенала трёх дисциплин – теории алгоритмов (в её прикладном аспекте), математической логики и теории информации. Важнейшим из таких средств стало понятие алгоритма. В последние годы в педагогику вошло понятие «педагогическая технология», важным компонентом которого также является понятие алгоритма.

В связи с бурным усовершенствованием компьютеров получила новое воплощение идея программированного обучения: обучающие, тренажерные и контролирующие комплексы на базе компьютеров требуют, прежде всего, педагогически и методически выверенного алгоритмического обеспечения. Алгоритмы работы таких систем должны быть направлены на достижение максимального образовательного результата. Это возможно лишь при успешном симбиозе прикладной теории алгоритмов и методики обучения соответствующей дисциплины. Ещё большие требования к алгоритмическо-методическому обеспечению учебного процесса предъявляет набирающее силу дистанционное обучение на базе компьютерной сети Интернет.

За время учёбы в школе учащиеся осваивают приёмы выполнения большого количества алгоритмов в самых разных науках и, прежде всего, в математике. Сначала – алгоритмы выполнения четырёх арифметических действий над различными числами – натуральными, целыми, дробными, комплексными. Эти алгоритмы служат основой для многочисленных вычислительных или численных алгоритмов, которыми изобилуют все разделы математической науки. Это обусловлено тем, что к четырём арифметическим операциям могут быть сведены очень многие другие операции. Одни из таких алгоритмов приводят к точным результатам, другие – к приближенным, но в последнем случае позволяют достигать любой наперёд заданной степени точности. Много внимания в школьном курсе математики уделяется алгоритмам решения линейных и квадратных уравнений и их систем.

В вузовском курсе высшей математики к этим алгоритмам добавляются алгоритмы вычисления определителей различных порядков, решения систем линейных алгебраических уравнений (с любым числом неизвестных), вычисления суммы и произведения матриц, рангов матриц, неопределённых интегралов от рациональных функций и многие другие.

Что касается алгоритмов приближённых вычислений, то они, по существу, составляют основу всей прикладной математики. Простейшим из таких является, например, алгоритм извлечения корня квадратного, который позволяет находить корень из любого положительного числа приближённо, но с любой наперёд заданной точностью, с помощью последовательности делений, умножений и вычитаний. Сюда относятся алгоритмы приближённых вычислений значений функций посредством разложения этих функций в степенные ряды, алгоритмы приближённых вычислений определённых интегралов (методы прямоугольников, трапеций, Симпсона), методы решения дифференциальных уравнений и их систем и т.д.

Важную роль в математике играют геометрические алгоритмы. Первое строго логическое изложение геометрии дал Евклид (III век до Р. Х.) в своём знаменитом дошедшем до нас трактате «Начала», состоящем из 13 книг. Все доказательства, приводимые в них, – своего рода алгоритмы, по которым следует движение мысли, чтобы от начальных данных достичь требуемого результата. Но один раздел евклидова трактата можно, вне всякого сомнения, рассматривать как предтечу и понятия алгоритма, и проблематики теории алгоритмов, которая встала на повестку дня математической науки лишь двадцать три века спустя, в первой трети XX века. Это раздел – геометрические построения на плоскости с помощью циркуля и линейки.

Теория решения таких задач начинается с двух аксиом: аксиомы построения циркулем и аксиомы построения линейкой. Первая говорит о том, что с помощью циркуля можно построить окружность с центром в любой точке плоскости и радиуса равного любому отрезку, заданному на плоскости. Вторая говорит, что с помощью линейки через любые две различные точки плоскости можно провести единственную прямую (и никак иначе линейка использоваться не может). Затем, опираясь на эти аксиомы, решается ряд простейших задач на построение. Строятся: луч; отрезок; угол равный данному; биссектриса угла; точка, делящая отрезок пополам; перпендикуляр к данной прямой, проходящий через данную точку; прямая, проходящая через данную точку и параллельная данной прямой; треугольник по трём сторонам; треугольник по стороне и двум прилежащим к ней углам; треугольник по двум сторонам и углу между ними; прямоугольные треугольники по двум элементам; касательная к данной окружности, проходящая через данную точку. Решение каждой из этих задач даётся соответствующим алгоритмом.

Древнегреческие математики поставили и решили большое количество интересных задач на построение с помощью циркуля и линейки. Но некоторые такие задачи им всё же решить не удалось.

Отметим в заключение этого пункта, что известные отечественные специалисты в области математической логики и теории алгоритмов В.А. Успенский и А.Л. Семёнов необычайно высоко оценили роль алгоритмических концепций в процессе обучения и воспитания современного человека, полагая, что ее можно сравнить «лишь с ролью письменности» [5, с. 230].

Алгоритмы и компьютеры. Наконец, алгоритмы самым тесным образом связаны с компьютерами, поскольку основой всякой компьютерной программы на любом языке программирования является алгоритм решения задачи. Алгоритмы и их компьютерное выражение имеют многочисленные интересные приложения к исследованию математических гипотез в таких областях классической математики, как численный анализ, комбинаторный анализ, статистика, алгебра, теория чисел. В них компьютер оказывается полезным, а иногда и незаменимым помощником в выполнении технически сложных, но вполне поддающихся алгоритмизации и программированию рутинных работ. Ярким примером такого сотворчества человека и машины является решение знаменитых математических проблем – великой проблемы Ферма [6] и проблемы четырёх красок [7]. Более простым, но все же содержательным математическим задачам, решаемым с помощью компьютерных методов, посвящена книга [8].

При решении математических задач с помощью компьютеров последние не только помогают человеку в переборе огромного количества вариантов и не сводятся только к вычислительной проверке каких-то количественных аспектов математических гипотез. Попытки «механизировать» классическую математику имеют методологическое значение: они приводят к лучшему уяснению тех понятий, о которых, как нам казалось, мы знаем всё (до тех пор, пока нам не пришлось объяснять их машине).

Алгоритмы и логика. Алгоритмы имеют и ещё одно чрезвычайно важное значение для математики. Оно заключается в их методологическом характере – алгоритмы служат одним из средств обоснования математики. Дело в том, что составление алгоритма и основанной на нём компьютерной программы из некоторого множества базисных команд очень сходно с построением математического доказательства, исходя из заданного множества аксиом с использованием конечного набора правил вывода. На эту аналогию и на понятие алгоритма опирается строгое понятие логического исчисления, которое служит формальным уточнением интуитивных понятий «вывода» и «доказательства».

Таким образом, эти две математические теории – математическая логика и теория алгоритмов – находятся в теснейшем взаимодействии, которое особенно плодотворно в тех разделах математической логики, которые исследуют аксиоматические математические теории и методы доказательства математических теорем [9–17].

Такое понимание взаимосвязи между мышлением логическим и мышлением алгоритмическим стало основой для так называемого конструктивного логико-философского направления в обосновании математики, когда в начале XX века возникла необходимость преодоления кризисных явлений, создавшихся в основаниях математики. Такой подход предполагал полное выведение доказательств из области интуиции и превращение их в чёткую алгоритмическую последовательность ясных шагов, приводящую к конкретному конструируемому объекту. (Подробнее о кризисе в математике и путях выхода из него см.: [9; 10]).

От конкретных алгоритмов к науке об алгоритмах. В конце XIX века немецкий математик Д. Гильберт заметил, что математика есть искусство называть разные вещи одним и тем же именем. Интуитивное понимание алгоритма, в том или ином виде присутствующее в математике на протяжении всей многовековой истории её развития, несомненно, служит одним из подтверждений этой мысли знаменитого математика. Под алгоритмом всегда понимался некий единый общий метод решения бесконечного числа однотипных конкретных задач, составляющих так называемую массовую проблему, своего рода решение этой проблемы «в общем виде». Вслед за алгоритмами арифметических действий с числами и знаменитым алгоритмом Евклида математика наполнилась многочисленными алгоритмами из самых разнообразных её областей: геометрии, алгебры, математического анализа. Эти алгоритмы прочно вошли в математическую науку и её практические приложения.

Тем не менее ряд массовых проблем не находил алгоритмов для своего решения. Такими были, например, в древние времена три классические задачи древности на построение циркулем и линейкой (о трисекции угла, об удвоении куба и о квадратуре круга), а в новейшие времена – 10-я проблема Гильберта о разрешимости диофантовых уравнений. У математиков возникли подозрения, что эти и ещё ряд подобных массовых проблем не имеют алгоритмов для своего решения. Для строгого обоснования таких утверждений требовалось сделать доселе интуитивно понимаемое понятие алгоритма понятием строго математическим, сделать его объектом изучения математики.

К строгому анализу интуитивного понимания понятия алгоритма подвигали также идеи голландского математика Л. Э. Я. Брауэра и немецкого математика Г. Вейля, высказанные ими в 1920-е годы, в которых они обратили внимание на различие между конструктивными и неконструктивными доказательствами в математике. Рассуждая конструктивно при доказательстве существования какого-либо объекта с заданными свойствами, математик указывает способ (алгоритм) его получения.

Классическим примером теоремы, доказательство которой носит ярко выраженный конструктивный, алгоритмический, характер, является теорема Л. Эйлера о графах, решающая его задачу о кёнигсбергских мостах. Эта теорема дает критерий эйлеровости графа. Ее доказательство состоит в разработке конкретного алгоритма, который для любого графа, удовлетворяющего условию теоремы (все вершины графа должны быть четными) конструктивно строит объект, существование которого утверждается в заключении теоремы – эйлеров цикл в графе (указывает в этом графе путь, начинающийся и заканчивающийся в одной и той же вершине графа и проходящий точно один раз через каждую вершину графа). Здесь можно вспомнить и великого Г.-В. Лейбница, который ещё в XVII веке призывал к созданию исчисления, способного «идеи заменить вычислениями» [18, с. 497].

В отличие от этого, неконструктивное доказательство или, как говорят, доказательство чистого существования не позволяет явно построить такой объект; оно лишь устанавливает логическое противоречие, если предположить, что такого объекта не существует. Примерами доказательств такого вида служат доказательства следующих теорем: 1) об ограниченности непрерывной на отрезке функции (о существовании верхней и нижней границ непрерывной на отрезке функции); 2) о расширении всякого отношения порядка до линейного отношения порядка.

В итоге в 1930-е годы была создана и начала развиваться общая наука об алгоритмах – теория алгоритмов. В ней объектом исследования стало само понятие алгоритма как таковое и его самые общие свойства. Первое математическое уточнение (формализацию) понятия алгоритма предложили независимо друг от друга американский математик и логик Э. Л. Пост (1936) и английский математик и инженер А. М. Тьюринг (1937). Они описали точными математическими терминами некие идеализированные вычислительные машины. Понятия, введённые Постом и Тьюрингом, не отличались существенно друг от друга; впоследствии они получили название «машины Тьюринга».

Важным примером алгоритмической проблемы, к которой могут быть сведены другие алгоритмические проблемы, служит проблема вычисления значений функции. Функции, для которых такие алгоритмы существуют, называются (алгоритмически) вычислимыми. Здесь возникает вопрос, как охарактеризовать вычислимые функции. Такую характеристику в 1936 году дали американские математики А. Чёрч и С. К. Клини. Оказалось, что класс вычислимых по А. Чёрчу и С. К. Клини функций (названных частично рекурсивными) совпадает с классом функций, вычислимых на машинах Тьюринга. Это дало основание высказать знаменитый тезис Чёрча об адекватности строгого понятия частичной рекурсивности функции с интуитивным понятием её алгоритмической вычислимости.

В 1951 году советский математик А. А. Марков предложил ещё одно уточнение общего понятия алгоритма – «нормальные алгорифмы» (или алгоритмы), предназначенные для оперирования со словами в некотором алфавите. Оказалось, что и нормальные алгорифмы Маркова эквивалентны машинам Тьюринга в том смысле, что с их помощью вычислимы одни и те же функции.

Таким образом, откристаллизовалось основное содержание формально-логической теории алгоритмов – изучение принципиальных возможностей идеализированных вычислительных машин и свойств классов вычислимых с их помощью функций.

Построенные формализации понятия алгоритма и вычислимой функции позволили строго доказать алгоритмическую неразрешимость большого количества массовых проблем и начать развивать общую (или абстрактную) теорию алгоритмов, в которой конкретный формализм отошёл на второй план, а главными понятиями, по существу, снова стали интуитивно понимаемые понятия алгоритма и вычислимой функции.

Наконец, говоря о связи абстрактной теории алгоритмов с традиционной математикой, отметим, что эта теория выработала методы, позволившие в первой трети XX века доказать беспрецедентные теоремы об аксиоматическом методе, составляющем существо человеческого мышления в области математики. Это теоремы К. Гёделя о неполноте формальной арифметики и А. Тарского о невыразимости истинных утверждений формальной арифметики в языке самой формальной арифметики, А. Чёрча – о неразрешимости формализованного исчисления предикатов, А. Тарского – о разрешимости элементарной геометрии. Таким образом, абстрактная теория алгоритмов, развившись в XX веке как раздел математики, впитала в себя её методы и в то же время способствовала решению ряда исконно математических проблем.

Идеализированные вычислительные машины были введены задолго до появления реальных вычислительных машин. Тем не менее нет никаких сомнений в том, что идеология идеализированных вычислительных машин стала теоретической основой для программного обеспечения реальных вычислительных машин и во многом способствовала их скорейшему появлению. Когда же такие машины появились и стали бурно совершенствоваться, вслед за формально-логической теорией алгоритмов стали возникать и развиваться многочисленные теории различных конкретных алгоритмов, призванных решать те или иные конкретные математические задачи, являвшиеся, в основном, математическими моделями реальных научных и производственных задач. Важнейшим вопросом о конкретных алгоритмах стал вопрос об их практической выполнимости на реально имеющихся вычислительных машинах в реально допустимое время. В связи с этим в 1960–70-е годы возникла теория вычислительной сложности алгоритмов, призванная расклассифицировать массовые проблемы и алгоритмы по их сложности. Если в 1930-е годы главным вопросом теории алгоритмов было различие между алгоритмически разрешимыми и алгоритмически неразрешимыми проблемами, то в 1960-е годы глубокое философское и практическое звучание приобрело различие между быстро решаемыми и долго решаемыми алгоритмическими проблемами, а сама теория сложности вычислений и алгоритмов стала одной из центральных в теории алгоритмов, да и в (прикладной) математике в целом.

Именно здесь возникла нерешенная до сих пор важнейшая проблема прикладной математики о взаимоотношении классов $\boldsymbol{P}$ и $\boldsymbol{NP}$ массовых проблем. Приблизительно говоря, к первому классу относятся массовые проблемы, допускающие для своего решения алгоритмы полиномиальной временной сложности, а ко второму – не имеющие таких алгоритмов, т. е. проблемы, решаемые алгоритмами только экспоненциальной временной сложности. Ясно, что $\boldsymbol{P} \subseteq \boldsymbol{NP}$. Проблема состоит в том, совпадают ли эти классы, или второй существенно шире первого.

Абстрактная теория алгоритмов, математическая логика и парадигмы программирования. Теория алгоритмов оказала существенное влияние на процессы создания компьютеров и программного обеспечения к ним. В теории алгоритмов были предугаданы основные концепции, заложенные в аппаратуру и языки программирования компьютеров. Различные формализации понятия алгоритма, выработанные абстрактной теорией алгоритмов, стали основами парадигм программирования и соответствующих алгоритмических языков программирования, созданных на основе этих парадигм. Так, машины Тьюринга привели к императивному программированию (языки программирования Бейсик, Фортран, Паскаль), нормальные алгоритмы Маркова – к продукционному программированию (языки программирования символьной обработки информации COMIT, SNOBOL, Рефал, ПРОЛОГ), теория рекурсивных функций – к функциональному программированию (язык программирования Лисп), теория акторов – к объектно-ориентированному программированию (языки программирования Смолток, С++) [19].

Теория алгоритмов развивалась в тесном взаимодействии с математической логикой. Это, вне всякого сомнения, обусловлено тесной взаимосвязью алгоритмического и логического мышления человека, схожестью алгоритмов с процессами построения логических умозаключений, использованием обеими дисциплинами формальных языков. В результате этого взаимодействия откристаллизовалась следующая методологическая цепочка, которая должна быть уяснена каждым будущим специалистом в области информатики, вычислительной техники, информационных систем и технологий, компьютерной безопасности:

МАТЕМАТИЧЕСКАЯ ЛОГИКА [Формальные языки] ⇒

ТЕОРИЯ АЛГОРИТМОВ (МАТЕМАТИЧЕСКИЕ МОДЕЛИ ВЫЧИСЛИМОСТИ) [Алгоритмические языки] ⇒

ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ [Языки программирования].

Теория алгоритмов наряду с неотделимой от нее математической логикой являются фундаментом, на котором зиждутся теория и практика компьютеров, программирования и информатики. Ведь каждая компьютерная программа представляет собой выражение алгоритма решения некоей задачи на одном из алгоритмических языков, которые тесно связаны с формализованными языками математической логики.

Теория алгоритмов как учебный предмет. Изложенные представления об историческом развитии, предмете, содержании и методах теории алгоритмов как научной дисциплины позволяют высказать следующие методические соображения относительно того, как должен быть организован соответствующий учебный предмет «Теория алгоритмов».

Во-первых, курс теории алгоритмов должен непосредственно следовать за курсом математической логики [9–13]. Далее теория алгоритмов как учебный предмет должна представлять науку об алгоритмах в двух её ипостасях – как теория конкретных алгоритмов (или интуитивно-содержательная теория алгоритмов) и как формально-логическая (абстрактная) теория алгоритмов. Такое построение курса будет наилучшим образом отвечать одному из положений принципа развивающего обучения – положению о соответствии характеру исторического развития научной дисциплины.

Говоря о теории конкретных алгоритмов, необходимо сначала охарактеризовать исторически сложившееся неформальное (интуитивное) понятие алгоритма, описать два языка представления таких алгоритмов: язык блок-схем и алгоритмический язык. Рассмотреть разнообразные примеры конкретных алгоритмов. Среди них алгоритмы, решающие так называемые игры двух лиц, в частности, игру «n предметов». С алгоритмической точки зрения анализируется игра «Ханойская башня», приводится алгоритм К.-Ф. Гаусса определения даты Святой Пасхи – Светлого Христова Воскресения. Обсудить вопросы доказательства корректности (правильности) алгоритмов.

Далее на неформальном (интуитивно-содержательном) уровне можно обсудить вопросы анализа сложности алгоритмов и массовых проблем, рассмотрев примеры анализа на сложность нескольких простых алгоритмов – суммирования членов последовательности, вычисления степени и факториала, вычисления значения многочлена по схеме Горнера, алгоритмов сложения и умножения матриц. Затем можно проанализировать на сложность алгоритмы, решающие ряд практически важных массовых проблем на графах. Среди них теорема об эйлеровых графах, о которой говорилось выше, проблема гамильтоновости графа, алгоритмы поиска путей в лабиринтах (в частности, алгоритм Тесея), алгоритм Дейкстры поиска кратчайшего пути в сети, алгоритм Форда-Фалкерсона о максимальном потоке в сети. Наконец, здесь нужно показать, каковы способы сравнения и классификации массовых проблем по их сложности и выделить различные классы сложности массовых проблем, важнейшими из которых являются классы ***P, NP*** и ***NP***-полных массовых проблем. Разъяснить существо одной из важнейших нерешенных современных проблем информатики о совпадении или несовпадении классов ***P*** и ***NP.***

Далее следует приступить к формально-логической (абстрактной) теории алгоритмов и, прежде всего, описать процедуру перехода от конкретных алгоритмов, примеры которых рассматривались выше, к общей теории алгоритмов. Затем охарактеризовать формализации понятий алгоритма и вычислимой функции – машины Тьюринга и рекурсивные функции, установить их логическую равносильность, сформулировать тезисы Тьюринга и Чёрча.

При боле глубоком изучении курса могут быть рассмотрены основные методы общей теории алгоритмов: метод нумераций и метод диагонализации (или диагональный метод Кантора). С их помощью устанавливаются фундаментальные результаты общей теории алгоритмов, доказанные С. Клини, – теорема о параметризации (*s-m-n*-теорема), теорема об универсальной функции и теорема о неподвижной точке; обсуждаются применения этих теорем в теории и практике программирования. Здесь же можно рассмотреть ряд вопросов алгоритмической теории множеств (разрешимым и перечислимым множествам), в которой два классических способа задания множеств изучаются с точки зрения алгоритмической эффективности, конструктивности. Важность понятий разрешимости и перечислимости множеств для оснований математики связана с тем, что язык теории множеств является в известном смысле универсальным языком математики.

Апофеозом курса теории алгоритмов являются вопросы, связанные с алгоритмически неразрешимыми массовыми проблемами – в теории алгоритмов и в математике. Среди них – пример функции, не вычислимой ни на какой машине Тьюринга, проблемы распознавания самоприменимости и распознавания применимости (проблема остановки) машин Тьюринга, проблемы распознавания нулевых функций и равенства двух вычислимых функций, проблемы входа и выхода, проблема распознавания общерекурсивных (всюду определенных) функций. В связи с алгоритмической теорией множеств может быть доказано существование перечислимых, но не разрешимых множеств, а также существование неперечислимых множеств.

Важнейшее методологическое значение для информатики и программирования имеет фундаментальная теорема Райса об алгоритмической неразрешимости проблемы распознавания нетривиальных свойств вычислимых функций. Эта теорема, по сути, означает, что не существует единого алгоритма, который для каждой вычислимой функции определял бы, обладает ли эта функция тем или иным свойством или нет, например, является ли эта функция постоянной, монотонной, периодической, ограниченной и т.п. Но это лишь первое приближение к пониманию смысла этой теоремы. Дело в том, что мы пытаемся создать единый алгоритм, который имеет дело с функциями. Но что значит иметь дело с функцией? Функция должна быть как-то задана. С точки зрения теории алгоритмов, функция задается вычисляющим ее алгоритмом. При этом ясно, что каждая функция может вычисляться множеством алгоритмов. Разыскиваемый нами единый алгоритм как раз и имеет дело с алгоритмами, вычисляющими функции. Смысл теоремы Райса состоит в том, что *по описанию алгоритма, вычисляющего функцию, ничего нельзя узнать о свойствах функции, которую он вычисляет.* Еще раз подчеркнем – не существует единого алгоритма, применимого к описаниям всех вычисляющих алгоритмов всех функций. В частности, оказывается неразрешимой проблема *эквивалентности алгоритмов: по двум заданным алгоритмам нельзя узнать, вычисляют они одну и ту же функцию или нет.*

Каждый, кто имел дело с программированием (написанием ком-пьютерных программ), знает, что по тексту сколько-нибудь сложной программы, не запуская ее в работу, трудно понять, что она делает (какую функцию вычисляет). Если это понимание и приходит, то каждый раз по-своему; единого метода здесь не существует. Это своего рода практическое проявление теоремы Райса.

В курсе математической логики обсуждаются проблемы синтаксиса и семантики языка при рассмотрении формальных аксиоматических математических теорий (см. [9, раздел 7.4]). В теории алгоритмов появляется еще один аспект этой проблемы. Синтаксические свойства алгоритма – это свойства описывающих его текстов, т.е. свойства конечных слов в фиксированном алфавите. Семантические (или смысловые) свойства алгоритма связаны с тем, что делает алгоритм, что вычисляет; эти свойства естественно описывать в терминах функций, вычисляемых алгоритмами. Хорошо известно, что в процессе отладки программ синтаксические ошибки отыскиваются довольно легко (этому, в частности, способствуют и дополнительные программы-алгоритмы). Главные неприятности связаны именно с анализом семантики неотлаженной программы, т.е. с попытками установить, что же она делает вместо того, чтобы делать то, что мы хотим (и здесь нам уже никакие дополнительные программы помочь не могут). Образно выражаясь, можно сказать, что теорема Райса звучит так: *по синтаксису алгоритма ничего нельзя узнать о его семантике, точнее, не существует единого алгоритма, позволяющего узнавать это единым способом для всех алгоритмов.* Таким образом, теорема Райса осуществляет отделение синтаксиса алгоритма от его семантики.

С теоретической точки зрения алгоритмическая неразрешимость массовой проблемы – не неудача, а научный факт. Он означает лишь отсутствие единого способа для решения всех единичных задач данной бесконечной серии, в то время как каждая индивидуальная задача серии вполне может быть решена своим индивидуальным способом. Более того, может оказаться разрешимой (своим индивидуальным методом) не только каждая отдельная задача этого класса, но и целые подклассы задач этого класса, поэтому, если проблема неразрешима в общем случае, нужно искать ее разрешимые частные случаи. Задача в более общей постановке имеет больше шансов оказаться неразрешимой. Понимание сути проблемы алгоритмической неразрешимости и знание основных алгоритмических неразрешимостей является одним из важных элементов современной математической и логической культуры особенно для тех, кто имеет профессиональные дела, связанные с компьютерами, программированием, информатикой.

В соответствии с изложенной методической концепцией автором разработаны, написаны и изданы учебники по курсу теории алгоритмов для студентов учреждений высшего и среднего профессионального образования, а также высшего образования, выпущенные издательством ИНФРА-М [14] и издательским центром «Академия» [15]. (Их своего рода предтечей стало пособие [16]). В конце учебника [15] приведён обширный список литературы, разделённый по темам. Её можно использовать для более глубокого изучения вопросов, затронутых в этом пособии. В частности, в учебнике [14] вопросы формально-логической (абстрактной) теории алгоритмов рассматриваются более обстоятельно и доказательно. К числу формализаций понятия алгоритма добавляются нормальные алгоритмы А. А. Маркова, доказывается эквивалентность трех формализаций понятия алгоритма, о которых говорилось выше. Формулируется принцип нормализации Маркова. Более строго излагается теория сложности вычислений и массовых проблем. Доказываются теоремы К. Гёделя о неполноте формальной арифметики, А. Чёрча о неразрешимости формализованного исчисления предикатов (с использованием теории машин Тьюринга).

При изучении курса теории алгоритмов рекомендуются сборники задач [20; 21], разработанные автором на основе перечисленных принципов их построения.

Выводы. Таким образом, владение основами общей теории алгоритмов является непременным атрибутом фундаментальной образованности всякого современного высококвалифицированного специалиста в области IT-технологий (компьютеров, программирования и информатики). Теория алгоритмов развивалась в тесном взаимодействии с математической логикой. И изучается данный курс обычно после курса математической логики. Это, вне всякого сомнения, обусловлено тесной взаимосвязью алгоритмического и логического мышления человека, схожестью алгоритмов с процессами построения логических умозаключений и доказательств, использованием обеими дисциплинами формальных языков.

В соответствии с изложенными рекомендациями, преподаватель может сформировать содержание курса теории алгоритмов в зависимости от того, будущим специалистам какого уровня данный курс преподается – на уровне бакалавриата или магистратуры в системе высшего образования, или в системе среднего профессионального образования.

В заключение обратим внимание на следующее методологическое обстоятельство. Если XVIII и XIX века были веками господства в приложениях методов непрерывной математики, то XX век несомненно выдвинул на передний план в сфере приложений методы дискретной математики, к которой относятся теория алгоритмов и математическая логика. Это вызвано, прежде всего, гигантским наплывом информации и созданием компьютеров и компьютерных методов ее обработки. XXI век продолжает усиливать эту тенденцию. Дискретность тесно связана с алгоритмами; это одно из определяющих их свойств. Несбалансированное теорией безудержное овладение только алгоритмами формирует конструктивный характер поведения и как следствие конструктивный образ мыслей. Итак,

ДИСКРЕТНОСТЬ ⇒ АЛГОРИТМИЗМ ⇒ КОНСТРУКТИВИЗМ

– вот «знамена» XXI века. Куда же мы идем под этими знаменами? Если «непрерывный» характер мышления учил рассуждениям и логике и способствовал формированию личностей мыслящих и творящих, то «дискретный» образ мыслей учит действиям (нажимать кнопки компьютеров, гаджетов, телефонов, плееров, бытовой техники, решать только типовые задачи по образцам-алгоритмам и т. п.) и соответственным действенным компетенциям, способствуя формированию исполнителей и потребителей. Таков объективный ход вещей.

Такая тенденция современного общественного развития требует организации системы образования, которая могла бы нейтрализовать пагубное влияние дискретных процессов на подрастающие поколения и направить формирование будущих личностей на путь созидания и творчества, но это тема уже другого исследования.

СПИСОК ЛИТЕРАТУРЫ

1. **Игошин В. И.** Курс математической логики в системе среднего профессионального образования // Профессиональное образование в современном мире. 2017. Т. 7, №2. С. 1018–1022. DOI: 10.153/ PEMW20170211.
2. **Teaching** and Learning Discrete Mathematics Worldwide: Curriculum and Research, ICME-13. Monographs / Hart E. W. and Sandefur J. (eds.). 2018, 276 p. DOI: https://doi.org/10.1007/978-3-319-70308-4.
3. **Igoshin V. I.** Mathematics and Logic: Their Relationship in the Teaching of Mathematics // Teaching and Learning Discrete Mathematics Worldwide: Curriculum and Research, ICME-13. Monographs / Hart E. W. and Sandefur J. (eds.). 2018, pp. 253–271. DOI: https://doi.org/10.1007/978-3-319-70308-4_16.
4. **Бескин Н. М.** Роль задач в преподавании математики // Математика в школе. 1992. №4–5. С. 3–5.
5. **Успенский В. А., Семёнов А. Л.** Теория алгоритмов: основные открытия и приложения. М., 1987.
6. **Виолант-и-Хольц А.** Загадка Ферма. Трехвековой вызов математике: пер. с исп. М.: Де Агостини, 2014. 160 с.
7. **Альсина К.** Карты метро и нейронные сети. Теория графов: пер. с исп. М.: Де Агостини, 2014. 144 с.
8. **Нивергельт Ю., Фаррар Дж., Рейнгольд Э.** Машинный подход к решению математических задач: пер. с англ. М.: Мир, 1977.
9. **Игошин В. И.** Элементы математической логики: учебник для студ. учрежд. сред. проф. образования. 3-е изд., стер. М.: Академия, 2018. 320 с.
10. **Игошин В. И.** Математическая логика: учебное пособие. М.: ИНФРА-М, 2014. 399 с.
11. **Игошин В. И.** Математическая логика и теория алгоритмов: учеб. пособие для студ. высш. учеб. заведений. 4-е изд. М.: Академия, 2010. 448 с.
12. **Игошин В. И.** Логика с элементами математической логики. (Лекции для студентов гуманитарных специальностей). Саратов: Научная книга, 2004. 144 с.
13. **Игошин В. И.** Тетрадь по математической логике. Саратов: Наука, 2010. 64 с.
14. **Игошин В. И.** Теория алгоритмов: учебное пособие. М.: ИНФРА-М, 2017. 318 с.
15. **Игошин В. И.** Теория алгоритмов: учебное пособие для студ. учрежд. сред. проф. образования. М.: Академия, 2018. 320 с.
16. **Игошин В. И.** Основы теории алгоритмов. (Лекции для студентов, изучающих информатику и информационные технологии). Саратов: Наука, 2008. 96 с.
17. **Игошин В. И., Дубракова Л. А.** Начало курса «Теория алгоритмов» в обучении будущих учителей математики и информатики // Учитель – ученик: проблемы, поиски, находки: сб. научно-методич. трудов: Вып. 6. Саратов: Наука, 2008. С. 48–54.
18. **Лейбниц Г.-В.** Сочинения: в 4 т. М., 1984. Т. 3. С. 497.
19. **Игошин В. И.** Подготовка будущих учителей математики и информатики в области дисциплин дискретной математики в условиях бакалавриата и магистратуры // Образование и наука. 2013. №7 (106). С. 85–100.
20. **Игошин В. И.** Задачи и упражнения по математической логике и теории алгоритмов. М.: Академия, 2008. 304 с.
21. **Игошин В. И.** Сборник задач по математической логике и теории алгоритмов: учебное пособие. М.: КУРС: ИНФРА-М, 2017. 392 с.

REFERENCES

1. **Igoshin V. I.** [Module «mathematical logic» in the system of vocational training]. *Professionalnoe obrazovanie v sovremenom mire = Professional education in the modern world,* 2017, vol. 7, no. 2, pp. 1018–1022. DOI: 10.153/ PEMW20170211.
2. **Hart E. W., Sandefur J.** (eds.) *Teaching and Learning Discrete Mathematics Worldwide: Curriculum and Research,* ICME-13 Monographs, 2018, 276 p. https://doi.org/10.1007/978-3-319-70308-4.
3. **Igoshin V. I.** *Mathematics and Logic: Their Relationship in the Teaching of Mathematics. In: Hart E. W., Sandefur J. (eds.) Teaching and Learning Discrete Mathematics Worldwide: Curriculum and Research,* ICME-13 Monographs, 2018, pp. 253–271. DOI: https://doi.org/10.1007/978-3-319-70308-4_16.
4. **Beskin N. M.** [Problems in Teaching Mathematic]. *Matimatika v shkole = Mathematic in school,* 1992, no. 4–5, pp. 3–5. (In Russ.)
5. **Uspenskii V. A., Semenov A. L.** [Theory of algorithms: main discoveries and applications]. Moscow, 1987. (In Russ.)
6. **Violant-i-Holz A.** [The Ferma Riddle. A Three-centure Challenge to Mathematics.]. Moscow, 2014, 160 p. (In Russian)

7. **Alsina Claudi.** [Metro maps and neuron webs]. Moscow, 2014, 144 p. (In Russian)
8. Nievergelt J., Farrar J. C., Reingold E. M. [Computer Approaches to Mathematical Problems]. Moscow, 1977, 160 p. (In Russ.)
9. **Igoshin V.I.** [Elements of mathematical logic: Textbook for students]. Moscow, 2018, 3rd ed., 320 p. (In Russian)
10. **Igoshin V.I.** [*Mathematical logic: Textbook for students*]. Moscow, 2014, 399 p. (In Russ.)
11. **Igoshin V.I.** [Mathematical logic and theory of algorithms: Textbook for students]. Moscow, 2010, 448 p. (In Russ.)
12. **Igoshin V.I.** [Logic with elements of mathematical logic]. Saratov, 2004, 144 p. (In Russ.)
13. **Igoshin V.I.** [Notebook in Mathematical Logic]. Saratov, 2010, 64 p. (In Russ.)
14. **Igoshin V.I.** [Theory of algorithms: Textbook for students]. Moscow, 2017, 318 p. (In Russ.)
15. **Igoshin V.I.** [Theory of algorithms: Textbook for students]. Moscow, 2013, 2018, 3rd ed., 320 p. (In Russ.)
16. **Igoshin V.I.** [Foundations of Algorithms Theory (Lectures for students learning informatics and IT)]. Saratov, 2008, 96 p. (In Russ.)
17. **Igoshin V.I., Dubrakova L.A.** [Beginning of the course «Theory of algorithms»in the training of future teachers of mathematics and computer science]. *Uchitel – uchenik: problem, poiski, nakhodki. Sb. Nauchno-metodicheskikh trudov = Teacher – disciple: problems, searches, finds. Sat. scientific-methodical. labours',* 2008, no. 6, pp. 48–54. (In Russ.)
18. **Leibnic G.-V.** [Works: in 4 vol.]. Moscow, 1984, vol. 3, p. 497. (In Russ.)
19. **Igoshin V.I.** [Bachelors and post-graduated education of mathematics and infirmatics teachers in discrete mathematical science]. *Obrazovanie i nauka = Education and science,* 2013, no. 7 (106), pp. 85–100. (In Russ.)
20. **Igoshin V.I.** [Problems and exercises on mathematical logic and theory of algorithms: Textbook for students]. Moscow, 2008, 304 p. (In Russ.)
21. **Igoshin V.I.** [Collection of problems on mathematical logic and theory of algorithms: Textbook for students]. Moscow, 2017, 392 p. (In Russ.)

Информация об авторах

Игошин Владимир Иванович – доктор педагогических наук, кандидат физико-математических наук, профессор, профессор механико-математического факультета, Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского (Российская Федерация, 410 012, Саратов, ул. Астраханская, 83, e-mail: igoshinvi@mail.ru)

Принята редакцией: 3.12.18

Information about the authors

Vladimir I. Igoshin – doctor of pedagogical sciences, candidate of physical and mathematical sciences, Professor, Department of Mathematics and Mechanics, Saratov State University (83 Astrakhanskaya Str., Saratov, 410012, Russian Federation, igoshinvi@mail.ru)

Received: December 3, 2018.