

ВСЕРОССИЙСКАЯ ОЛИМПИАДА ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ

В.М. Кирюхин

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по разработке заданий для школьного и муниципального этапов
всероссийской олимпиады школьников по информатике
в 2010/2011 учебном году**

Москва 2010

ОГЛАВЛЕНИЕ

Введение	3
1. Порядок разработки олимпиадных задач	5
2. Общие требования к олимпиадным задачам	6
3. Формирование комплектов олимпиадных задач	7
4. Типы олимпиадных задач	8
4.1. Типы задач для 9 – 11 классов	8
4.2. Типы задач для 7 – 8 классов	10
4.3. Типы задач для 5 – 6 классов	11
5. Содержание олимпиадных задач	13
6. Примеры олимпиадных задач	25
7. Рекомендации по проверке и оцениванию решений задач	26
7.1. Методика проверки решений задач участников	27
7.2. Система оценивания решений участников	29
7.3. Технология проверки решений участников	31
Список рекомендуемой литературы	40

Введение

Настоящие методические рекомендации подготовлены центральной предметно-методической комиссией по информатике в соответствии с Положением о всероссийской олимпиаде школьников и направлены на помощь муниципальным и региональным предметно-методическим комиссиям по информатике в составлении заданий для школьного и муниципального этапов всероссийской олимпиады школьников по информатике в субъектах Российской Федерации.

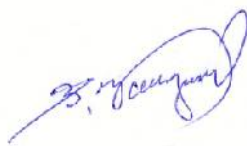
Олимпиадные задачи – это лицо любой олимпиады, и насколько квалифицированно они будут разработаны, зависит и успех соревнования в целом. Более того, хорошие задачи после олимпиады начинают жить своей жизнью: они достаточно быстро распространяются среди будущих участников олимпиад по информатике, их начинают активно использовать при подготовке к соревнованиям учителя и наставники, на базе них создаются новые олимпиадные задачи.

Подобрать или разработать хорошие задачи для школьного и муниципального этапов является достаточно сложной проблемой. Чтобы дать основные ориентиры разработчикам таких задач, центральной предметно-методической комиссией по информатике разработаны настоящие методические рекомендации, которые определяют требования к олимпиадным задачам и порядку их разработки, состав комплекта олимпиадных задач для каждого тура, типы и содержание задач. Кроме того, методические рекомендации содержат описание методики проверки решений задач и рекомендуемой системы оценивания, а также подходы к выбору средств автоматизации процесса проверки решений участников. Полезными будут также примеры олимпиадных задач и перечень ссылок на Интернет-ресурсы, а также список рекомендуемой литературы.

Центральная предметно-методическая комиссия по информатике выражает надежду, что представленные методические рекомендации окажутся полезными при составлении задач для школьного и муниципального этапов всероссийской олимпиады школьников по информатике, и желает успехов членам муниципальных и региональных предметно-методических комиссий в этом не простом деле. В случае необходимости, дополнительную информацию по представленным методическим материалам можно получить по электронной почте, обратившись по адресу support@rusolymp.ru в центральную предметно-методическую комиссию по информатике.

Настоящие методические рекомендации утверждены на заседании центральной предметно-методической комиссии по информатике (протокол № 7 от 24 июня 2010 года).

Председатель центральной
предметно-методической комиссии
по информатике



В.М. Кирюхин

1. Порядок разработки олимпиадных задач

В соответствии с п.п. 22 и 31 Положения о всероссийской олимпиаде школьников (далее – Олимпиада) школьный этап Олимпиады по информатике проводится по олимпиадным заданиям, разработанным муниципальной предметно-методической комиссией по информатике, а муниципальный этап – по олимпиадным заданиям, разработанным региональной предметно-методической комиссией по информатике.

В общем случае процесс создания олимпиадной задачи по информатике включает следующие основные этапы:

- 1) разработка идеи и подготовка текста условия задачи;
- 2) разработка требований к используемому в процессе решения задач программному обеспечению и к форме представления результатов решения задач;
- 3) разработка методики проверки решений задач, включая систему тестов для проверки решений в виде готовых к исполнению программ, написанных с использованием допустимых алгоритмических языков и систем программирования;
- 4) подготовка системы оценивания решений каждой задачи;
- 5) проверка корректности оценивания различных вариантов решений каждой задачи, включая частичные и полные решения;
- 6) разработка для каждой задачи дополнительного программного обеспечения, включая проверяющие программы, если предполагается использовать при проверке решений участников специализированные программные системы автоматической проверки решений участников.

Вполне очевидно, что процесс создания олимпиадной задачи является итерационным. Очень часто случается, что условие задачи может кардинально измениться в зависимости от результатов выполнения последующих этапов.

Результатом разработки соответствующими предметно-методическими комиссиями олимпиадных задач для школьного и муниципального этапа Олимпиады является комплект материалов, включающий:

- тексты олимпиадных задач;
- методику проверки решений задач, включая при необходимости комплекты тестов в электронном виде;
- описание системы оценивания решений задач;
- методические рекомендации по разбору предложенных олимпиадных задач.

В случае необходимости, предметно-методическая комиссия соответствующего этапа предоставляет также дополнительные материалы, необходимые для автоматизированной проверки решений участников, включая проверяющие программы, позволяющие для каждой

задачи определять правильность полученного решения в автоматическом режиме. Кроме того, в этом случае предметно-методические комиссии должны также подготовить организаторам и жюри соответствующего этапа вариант Памятки участника.

Комплект названных материалов должен передаваться в оргкомитет соответствующего этапа Олимпиады не позднее 7 календарных дней до начала соревнования, чтобы оргкомитет и жюри имели возможность подготовить необходимую компьютерную технику и программное обеспечение для проведения туров и проверки решений участников. При этом ответственность за неразглашение текстов олимпиадных задач и системы оценивания их решений до начала соревнований лежит на оргкомитете этого этапа Олимпиады.

2. Общие требования к олимпиадным задачам

Для проведения школьного и муниципального этапов всероссийской олимпиады школьников по информатике могут использоваться как переработанные и дополненные задачи, ранее использованные на других олимпиадах по информатике, так и оригинальные задачи, разработанные муниципальными и региональными методическими комиссиями. Основными критериями отбора олимпиадных задач должны быть следующие показатели [15]:

- оригинальная формулировка задачи или оригинальная идея ее решения для конкретного состава участников олимпиады;
- в тексте условия задачи не должны встречаться термины и понятия, выходящие за пределы изучаемых в рамках базового учебного плана предметов; в крайних случаях, они должны быть определены или конкретизированы;
- задача должна быть однозначно определена, т.е. в ее формулировке не должно быть неоднозначностей, чтобы участник олимпиады решал именно ту задачу, которую задумали авторы;
- задача не должна требовать для своего решения специальных знаний;
- формулировка задачи должна предполагать наличие этапа формализации при ее решении, т.е. переход от неформальной постановки задачи к формальной;
- задача должна быть разумной сложности и трудоемкости.

Важной особенностью задач, используемых при проведении школьного и муниципального этапов, является ориентация их на проверку развития у школьников теоретического мышления, логики, а также творческих способностей и интуиции. Предлагаемые задачи должны предоставлять возможность школьникам без специальных знаний решать нестандартные и новые для них задачи. Каждая задача должна позволять участникам сделать для себя небольшое открытие и в полной мере раскрыть имеющийся у них творческий потенциал.

Особенно это важно для школьного этапа Олимпиады, основная цель которого – выявление наиболее талантливых школьников, начиная с 5–6 классов, и создание в дальнейшем всех необходимых условий для их творческого роста, например, путем привлечения в соответствующие кружки, образовательные организации системы дополнительного образования и т.п.

3. Формирование комплектов олимпиадных задач

При формировании комплектов задач для школьного и муниципального этапов Олимпиады следует учитывать возрастные особенности участников, связь предлагаемых задач с программами изучения информатики и математики в образовательных организациях конкретного муниципального образования или региона, а также тот факт, что целью проведения начальных этапов Олимпиады является выявление наиболее талантливых школьников, которые увлечены информатикой и вне школьной программы самостоятельно занимаются изучением информатики в рамках системы дополнительного образования или с родителями. Здесь важно также учитывать, что в школьном этапе Олимпиады могут принимать участие обучающиеся 5 – 11 классов, в то время как в муниципальном этапе – только обучающиеся 7 – 11 классов. Но это не значит, что наиболее талантливым школьникам 5 – 6 классов путь на следующие этапы закрыт. В частности, выступая на школьном этапе за 7 класс и завоевав право участвовать в муниципальном этапе, такие школьники могут быть допущены к этому этапу, оформив обучение в форме экстерната по предмету «Информатика и ИКТ» в 7 классе или выше. То же самое можно сказать и об участниках муниципального этапа из 7 – 8 классов, которые могут претендовать на участие в региональном этапе Олимпиады.

Учитывая вышесказанное, центральная предметно-методическая комиссия по информатике рекомендует для школьного этапа формировать разные комплекты задач для 5-6, 7-8 и 9-11 классов, а для муниципального этапа – разные комплекты задач для 7-8 и 9-11 классов. Количество задач в каждом комплекте должно быть не менее трех для каждого тура и определяется соответствующей предметно-методической комиссией по информатике.

Задачи в каждом комплекте должны быть такой сложности, чтобы дать возможность проявить себя как недостаточно подготовленным, так и сильным участникам. Здесь важно не отпугнуть сложностью задач только начинающих свой путь в олимпиадном движении учащихся, а вовлечь их в олимпиадное движение по информатике и усилить их мотивацию к дальнейшему совершенствованию своих знаний и умений. С другой стороны, и сильные участники должны иметь возможность в полной мере продемонстрировать свои творческие способности, чтобы по результатам их выступлений можно было выявить лучшего из них, причем желательно одного, а не многих.

Оценить сложность комплекта задач можно только по результатам выступления всех участников на основе распределения количества набранных баллов по участникам [15]. Здесь идеальным может быть вариант, когда кривая распределения количества набранных баллов по участникам совпала бы с прямой, проходящей от точки с максимально возможным количеством баллов и до нуля. Это говорило бы о том, что данный комплект задач оптимально продифференцировал всех участников по уровню их подготовки и творческим способностям и его сложность полностью соответствует уровню подготовки всех участников, в частности, половина участников набрала бы более половины от максимально возможного количества баллов.

Если сравнивать уровень сложности комплектов задач для школьного и муниципального этапов Олимпиады, то вполне очевидно, что для муниципального этапа сложность комплекта задач должна быть выше. Этот факт предполагает тесное взаимодействие муниципальных и региональных предметно-методических комиссий по информатике, чтобы не было перекосов в ту или иную сторону.

4. Типы олимпиадных задач

При выборе типа задач для школьного и муниципального этапов необходимо руководствоваться следующими соображениями. Во-первых, в процессе решения олимпиадной задачи участники обязательно должны использовать компьютер. Во-вторых, при принятом разделении комплектов задач (5-6, 7-8 и 9-11 классы для школьного этапа и 7-8 и 9-11 классы для муниципального этапа) типы задач в каждом из комплектов также могут быть разными.

4.1. Типы задач для 9 – 11 классов

По давно устоявшейся традиции олимпиадные задачи для 9 – 11 классов могут быть трех типов. К задачам первого типа относятся стандартные задачи, решением которых является программа, формирующая по заданному входному файлу выходной файл. Задачи второго типа являются интерактивными. Решением задач этого типа также является программа, однако, в отличие от задач первого типа, вместо чтения исходных данных из входного файла и записи результата в выходной файл эта программа должна обмениваться данными с другой программой, определенной в условии задачи. В задачах третьего типа, которые называются задачами с открытым входом, решением является не программа, как в задачах первого или второго типов, а файлы выходных данных, соответствующие заданным в условии задачи входным файлам.

Разные задачи можно решать с использованием разных языков программирования и систем программирования. Список допустимых языков и систем программирования

устанавливается предметно-методической комиссией по информатике соответствующего этапа до начала проведения олимпиады с учетом настоящих рекомендаций.

Для задач, в которых решением является программа, в тексте условия указывается максимальное время работы программы на каждом тесте и размер доступной программе памяти. В случае превышения установленных ограничений, тест должен считаться не пройденным. При этом указанные ограничения по памяти включают всю память, используемую программой, в том числе память под код программы, системные нужды и т.д.

Решения перечисленных выше типов задач должны сдаваться участниками школьного и муниципального этапов олимпиады на проверку только на электронном носителе. В зависимости от типа задачи ее решением может быть либо текст программы, написанной с использованием допустимых сред программирования (для стандартных и интерактивных задач), либо набор выходных файлов, соответствующих заданным входным файлам (для задач с открытым входом), о чем должно сообщаться в условии задачи.

Если решением задачи является программа и для проверки решений участников используется программная среда проведения соревнований, то ее компиляция в проверяющей системе осуществляется с помощью команды компиляции, соответствующей выбранному участником языку программирования. Таблица команд компиляции должна быть доведена до сведения всех участников перед началом каждого тура и размещена в памятке участнику.

Для задач, решением которых является программа, в тексте условия рекомендуется указывать максимальное время работы программы и размер доступной программе памяти. Временем работы программы считается суммарное время работы процесса на всех ядрах процессора. Память, используемая приложением, включает всю память, которая выделена процессу операционной системой, включая память кода и стек.

Для программ-решений рекомендуется также использовать следующие ограничения: размер файла с исходным текстом программы не должен превышать 256 Кбайт, а время компиляции программы должно быть не больше одной минуты.

Участникам школьного и муниципального этапов Олимпиады разрешается использование в решениях задач любых внешних модулей и заголовочных файлов, включенных в стандартную поставку соответствующего компилятора.

В решениях задач участникам запрещается:

- создание каталогов и временных файлов при работе программы;
- любое использование сетевых средств;
- любые другие действия, нарушающие работу проверяющей системы, если она используется.

Для задач с открытым входом формат выходных файлов должен полностью соответствовать описанным в условии задачи требованиям. При нарушении этих требований выходной файл на проверку не принимается.

Муниципальные и региональные предметно-методические комиссии по информатике с учетом типа олимпиадных задач, разработанных для соответствующего этапа Олимпиады, формируют требования к форме представления результатов решений задач участников, которые заблаговременно доводятся до сведения участников и должны быть отражены в Памятке участнику, подготавливаемой для жюри соответствующего этапа.

4.2. Типы задач для 7 – 8 классов

Для обучающихся 7 – 8 классов рекомендуется использовать такие же типы задач, какие приведены в разделе 4.1. Поэтому все, сказанное о типах задач для обучающихся 9 – 11 классов, справедливо и для типов задач для обучающихся 7 – 8 классов.

Формой представления результатов решения задач для обучающихся 7 – 8 классов должна быть либо программа, написанная с использованием определенных муниципальной или региональной предметно-методической комиссией по информатике языков и систем программирования, либо набор выходных данных, соответствующий заданному набору входных данных (для задач с открытым входом). Если решением задачи является программа, то допускается ввод данных либо из входного файла `input.txt`, либо из стандартного потока ввода, т.е. с клавиатуры, а вывод допускается как в выходной файл `output.txt`, так и в стандартный поток вывода, т.е. на экран монитора.

Рекомендуется при формировании комплекта задач для каждого тура включать в его состав задачи различного типа, чтобы дать возможность проявить свои знания и умения участникам с различным уровнем подготовки.

4.3. Типы задач для 5 – 6 классов

Если ранее рассмотренные типы задач характерны как для школьного, так и для муниципального этапа Олимпиады, то типы задач для 5 – 6 классов ориентированы только на проведение школьного этапа. К сожалению, во многих регионах категория обучающихся 5 – 6 классов практически не привлекается к участию в школьном этапе. Считается, что такие школьники не готовы к решению олимпиадных задач по информатике в силу того, что в соответствии с Базисным учебным планом предмет «Информатика и ИКТ» не входит в федеральный компонент для 5-6 классов, а относится к школьному компоненту, и во многих школах обучающиеся 5-6 классов не имеют возможности в той или иной форме изучать этот предмет. При этом следует учитывать, что этот предмет представлен в федеральном компоненте Базисного учебного плана для 3 и 4 классов начальной ступени обучения, то

есть, к 5 классу у учащихся уже сформированы элементы информационных знаний и умений.

Отечественный и зарубежный опыт олимпиадного движения по информатике показывает, что если талантливость ребенка в области информатики выявляется и поддерживается в начальной школе, и далее непрерывно развивается, то именно такие школьники впоследствии становятся абсолютными чемпионами России и завоевывают золотые медали на международной олимпиаде по информатике. Таких примеров, когда шестиклассники уже участвовали в заключительном этапе Олимпиады по информатике и добивались хороших результатов, можно привести много. Из зарубежного опыта ярким доказательством вышесказанного является участие белорусского школьника уже в международной олимпиаде по информатике с 5 класса, когда он завоевал серебряную медаль, а уже в 2009 году, будучи восьмиклассником, он стал абсолютным чемпионом мира, опередив всех старшеклассников мира.

Целесообразность вовлечения в школьный этап Олимпиады по информатике младших школьников поддерживается также новым Федеральным государственным образовательным стандартом начального общего образования, утвержденным Приказом Минобрнауки России от 6 октября 2009 г. № 373 (http://www.edu.ru/db-mon/mo/Data/d_09/m373.html), который вводится в действие с 1 января 2010 года. В частности, этим стандартом предусматривается изучение особо важных для олимпиадной ориентации школьников тем, включая алгоритмы, множества, элементы комбинаторики, введение в понятие моделирования, начала логики, знакомство с информационными структурами, а также использование исполнителей для реализации алгоритмов. Особое внимание обращается на освоение младшими школьниками правил клавиатурного ввода, графического интерфейса, работы на компьютере и в компьютерной сети.

Понятно, что предъявлять к задачам для обучающихся 5 – 6 классов такие же требования, как и к задачам для старшеклассников, о которых речь шла выше, не совсем оправдано. Однако опыт проведения школьного этапа для таких школьников в ряде регионов страны показывает, что для выявления ранней одаренности у школьников младших классов могут с успехом использоваться следующие типы олимпиадных задач:

- задачи с упрощенными исполнителями;
- лабиринтные задачи;
- конечные клеточные игры, включая игры, основанные на шахматных сюжетах;
- задачи на геометрические построения;
- задачи на перестановки, сортировки, перекладывания, взвешивания, переправы;
- задачи типа «черный ящик», включая задачи на выявление закономерностей;

– задачи на тестирование заданных программ.

Понятно, что все олимпиадные задачи должны быть основаны на разработке алгоритма ее решения и реализации решения на компьютере. Однако формы представления результатов решения задачи могут быть разные. Самой простой формой является представление результатов решения задачи на бумажном носителе. Однако такой бумажный вариант не учитывает второй олимпиадной составляющей задачи по информатике – умения использовать компьютер для ее решения. Бумажный вариант представления алгоритма решения не характерен для олимпиадной информатики также в силу проблем, возникающих при проверке решений в таком виде, и непривлекательности для участников, поскольку в этом случае теряется грань между информатикой и математикой и возможности использования учащимся компьютера как партнера для проверки своего варианта решения, его исправления в случае ошибок, пошагового приближения к поиску оптимального решения задачи.

Заменой бумажной формы представления результатов решения олимпиадных задач для обучающихся 5 – 6 классов является запись решения в форме программы, предполагаемой достаточно распространенными программными системами учебного назначения, которые вполне доступны для младших школьников. Такими системами являются, например, КуМир, Скретч, Роботландия, различные вариации Лого и т.п. Использование этих систем на школьном этапе Олимпиады позволяет школьникам применить на практике возможности компьютера при решении задачи, представлять результаты своего труда на формальном языке, использовать элементы моделирования в процессе решения задачи и продемонстрировать свои умения работать с компьютером. Кроме того, в этом случае у жюри школьного этапа появляется возможность автоматизировать процесс проверки решений задач, что немаловажно при проведении олимпиады любого уровня.

Не следует исключать при проведении школьного этапа для обучающихся 5-6 классов возможность представления решений задач в виде файлов с текстом программы, написанной с использованием допустимых языков и сред программирования, о чем речь шла в разделе 4.2. Не исключено, что некоторые школьники младших классов уже могут программировать, и на Олимпиаде было бы неправильно не оценить их умения по достоинству.

5. Содержание олимпиадных задач

При определении содержания задач для школьного и муниципального этапов Олимпиады по информатике следует руководствоваться примерной программой по олимпиадной информатике, приведенной в книге [15]. Данная программа разработана с учетом Государственного образовательного стандарта по предмету «Информатика и ИКТ» (Приказ Минобразования 2004 года и дополнение к Приказу Минобрнауки России 2005

года) с перспективой введения стандарта второго поколения для всех ступеней школьного образования: начальной пропедевтической (3-6 классы), основной (7-8 классы), старшей предпрофильной (9 класс) и профильной (10-11 классы), а также на основе анализа структуры современного содержания олимпиад по информатике.

Программа является примерной, она отражает постоянно растущие требования к участникам Олимпиады в освоении наиболее важных разделов информатики с учетом развития олимпиадного движения, и обобщает 20-летний опыт развития содержания курса школьной информатики, банка задач региональных и заключительных этапов всероссийской олимпиады школьников, разработанных центральной предметно-методической комиссией по информатике.

Представленная ниже примерная программа по олимпиадной информатике содержит восемь разделов, которые раскрываются входящими в них темами. Каждая тема, в свою очередь, содержит дидактические единицы, более подробно раскрывающие ключевые знания и умения, на которые могут ориентироваться разработчики задач школьного и муниципального этапов Олимпиады по информатике.

Чтобы отразить в программе уровни сложности, каждая дидактическая единица в ней, характерная для участия в различных этапах всероссийской олимпиады школьников по информатике, имеет различное обозначение. В частности, выделено три уровня сложности – для 5-6, 7-8 и 9-11 классов, каждый из которых отмечен следующим образом:

- дидактическая единица *без* символа «*» означает, что она относится к *начальному* уровню сложности для учащихся 5-6 классов, и знание этих дидактических единиц позволяет учащимся впервые попробовать свои силы и определить свой олимпиадный уровень при участии в школьном этапе Олимпиады, обеспечивает достижение *понятийного* уровня требований к участнику олимпиад по информатике, позволяет осмысленно подойти к решению олимпиадных заданий;
- дидактическая единица *с одним* символом «*» соответствует *основному* уровню сложности для 7-8 классов, и знание этих дидактических единиц позволяет учащимся проявить свой творческий потенциал при участии в школьном и муниципальном этапах Олимпиады, обеспечивает достижение *продуктивного* уровня требований к участнику олимпиад по информатике, позволяет подойти к поиску оптимальных решений олимпиадных заданий и обеспечивает им возможность технологично представлять свои идеи;
- символы «**» означают, что дополнительное изучение этих дидактических единиц формирует у школьников устойчивые профильные умения в области олимпиадной подготовки для учащихся 9-11 классов, открывает перед участником олимпиадного состязания возможность проявить свой творческий потенциал на высоком уровне

представления решений олимпиадных заданий и позволяет сформировать портфолио достижений такого учащегося на уровне дипломов победителей и призеров региональных и заключительных этапов всероссийской олимпиады школьников по информатике.

С учетом сказанного примерная программа по олимпиадной информатике представляет собой следующее.

1. Математические основы информатики

1.1. Функции, отношения и множества

1.1.1. Функции

1.1.2. Отношения (рефлексивность, симметричность, транзитивность, эквивалентность)

1.1.3. Множества (диаграммы Венна, дополнения)

*1.1.4. Обратная функция, композиция **

*1.1.5. Лексикографический порядок **

*1.1.6. Декартовы произведения **

*1.1.7. Вполне упорядоченные множества ***

*1.1.8. Мощность и счетность множества. Конечные и бесконечные множества ***

1.2. Основные геометрические понятия

1.2.1. Точка, прямая, отрезок, вектор, угол

1.2.2. Треугольник, прямоугольник, многоугольник

1.2.3. Выпуклые многоугольники

1.2.4. Декартовы координаты в евклидовом пространстве

*1.2.5. Евклидово расстояние **

*1.2.6. Векторное и скалярное произведение на плоскости **

1.3. Основы логики

1.3.1. Логические переменные, операции, выражения

1.3.2. Таблицы истинности

1.3.3. Булевы функции

*1.3.4. Формы задания и синтез логических функций **

*1.3.5. Преобразование логических выражений **

*1.3.6. Минимизация булевых функций ***

*1.3.7. Основные законы логики суждений ***

*1.3.8. Логика предикатов ***

1.4. Основы вычислений

1.4.1. Основы вычислений:

- *Правила суммы и произведения*
 - *Арифметические и геометрические прогрессии **
 - *Числа Фибоначчи **
 - *Принцип «включения-выключения» ***
- 1.4.2. *Рекуррентные соотношения **
- 1.4.3. *Матрицы и действия над ними ***
- 1.5. Методы доказательства
- 1.5.1. *Прямые доказательства*
- 1.5.2. *Доказательство методом «от противного»*
- 1.5.3. *Доказательство методом исключения*
- 1.5.4. *Доказательство через контрпример*
- 1.5.5. *Математическая индукция **
- 1.5.6. *Структура формальных доказательств ***
- 1.6. Основы теории чисел
- 1.6.1. *Простые числа*
- 1.6.2. *Деление с остатком*
- 1.6.3. *Наибольший общий делитель*
- 1.6.4. *Основная теорема арифметики **
- 1.6.5. *Взаимно простые числа **
- 1.6.6. *Делимость. Кольцо вычетов по модулю ***
- 1.7. Основы алгебры
- 1.7.1. *Многочлены и операции над ними. Решение квадратных уравнений. Теорема Виета **
- 1.7.2. *Общий случай теоремы Виета. Симметрические многочлены ***
- 1.7.3. *Понятие группы ***
- 1.7.4. *Теоремы о гомоморфизме и изоморфизме ***
- 1.8. Основы комбинаторики
- 1.8.1. *Перестановки, размещения и сочетания:*
- *Основные определения*
 - *Тождество Паскаля **
 - *Биномиальная теорема **
- 1.8.2. *Коды Грея: подмножества, сочетания, перестановки ***
- 1.8.3. *Таблицы инверсий перестановок ***
- 1.8.4. *Разбиения на подмножества. Числа Стирлинга ***

1.8.5. *Скобочные последовательности* **

1.9. Теория графов

1.9.1. *Типы графов*

1.9.2. *Маршруты и связность*

1.9.3. *Деревья*

1.9.4. *Операции над графами* *

1.9.5. *Остовные деревья* *

1.9.6. *Раскраска графов* *

1.9.7. *Эйлеровы и гамильтоновы графы* *

1.9.8. *Покрытия и независимость* **

1.9.9. *Укладка графов. Плоские (планарные) графы* **

1.9.10. *Двусвязность графа. Мосты, блоки, точки сочленения* **

1.9.11. *Связь ориентированных ациклических графов и отношений порядка. Транзитивное замыкание* **

1.9.12. *Двудольные графы* **

1.9.13. *Потоки и сети* **

1.10. Основы теории синтаксического анализа

1.10.1. *Обратная польская запись*

1.10.2. *Синтаксический анализ простых выражений* *

1.10.3. *Регулярные выражения, конечные автоматы* **

1.11. Основы теории вероятностей

1.11.1. *Понятие вероятности и математического ожидания* *

1.11.2. *Аксиомы теории вероятностей* **

1.11.3. *Основы вычисления вероятностей* **

1.12. Основы теории игр

1.12.1. *Понятие игры и результата игры*

1.12.2. *Простейшие игры*

1.12.3. *Простейшие стратегии игры* *

1.12.4. *Игры на матрицах* **

1.12.5. *Решение игровых задач с использованием функции Гранди* **

2. Разработка и анализ алгоритмов

2.1. Алгоритмы и их свойства

2.1.1. *Понятие алгоритма*

2.1.2. *Концепции и свойства алгоритмов*

2.1.3. *Запись алгоритма на неформальном языке*

2.2. Структуры данных

2.2.1. *Простые базовые структуры*

2.2.2. *Множества*

2.2.3. *Последовательности*

2.2.4. *Списки*

2.2.5. *Неориентированные графы **

2.2.6. *Ориентированные графы **

2.2.7. *Деревья **

2.2.8. *Пирамида и дерево отрезков ***

2.2.9. *Сбалансированные деревья ***

2.2.10. *Хэш-таблицы и ассоциативные массивы ***

2.2.11. *Бор ***

2.3. Основы анализа алгоритмов

2.3.1. *Нотация O большое **

2.3.2. *Стандартные классы сложности **

2.3.3. *Асимптотический анализ поведения алгоритмов в среднем и крайних случаях **

2.3.4. *Компромисс между временем и объемом памяти в алгоритмах ***

2.3.5. *Использование рекуррентных отношений для анализа рекурсивных алгоритмов ***

2.3.6. *NP-полнота ***

2.4. Алгоритмические стратегии

2.4.1. *Алгоритмы полного перебора*

2.4.2. *"Жадные" алгоритмы **

2.4.3. *Алгоритмы "разделяй и властвуй" **

2.4.4. *Перебор с возвратом **

2.4.5. *Эвристики ***

2.5. Рекурсия

2.5.1. *Понятие рекурсии*

2.5.2. *Рекурсивные математические функции **

2.5.3. *Простые рекурсивные процедуры **

2.5.4. *Реализация рекурсии **

2.5.5. *Рекурсивный перебор с возвратами ***

2.6. Фундаментальные вычислительные алгоритмы

2.6.1. *Простые численные алгоритмы*

2.6.2. *Классические комбинаторные алгоритмы*

- 2.6.3. *Алгоритмы с подмножествами: генерация, восстановление по номеру и построение номера, генерация следующего и предыдущего (прибавление и вычитание единицы)*
- 2.6.4. *Алгоритмы последовательного и бинарного поиска*
- 2.6.5. *Алгоритмы с сочетаниями и перестановками (генерация, восстановление по номеру и построение номера, генерация следующего и предыдущего) **
- 2.6.6. *Квадратичные методы сортировки (сортировка методом выбора, сортировка вставками) **
- 2.6.7. *Сортировка подсчетом за линейное время **
- 2.6.8. *Алгоритмы сортировки за время $O(N \log N)$ (быстрая сортировка, пирамидальная сортировка, сортировка слиянием) ***
- 2.6.9. *Цифровая сортировка ***
- 2.6.10. *Алгоритм вычисления номера слова в лексикографически упорядоченном множестве перестановок его символов ***
- 2.6.11. *Арифметика многоразрядных целых чисел ***
- 2.7. Числовые алгоритмы
 - 2.7.1. *Разложение числа на простые множители*
 - 2.7.2. *Решето Эратосфена **
 - 2.7.3. *Алгоритм Евклида **
 - 2.7.4. *Расширенный алгоритм Евклида. Способы реализации алгоритма без деления ***
 - 2.7.5. *Решение линейных сравнений с помощью алгоритма Евклида ***
 - 2.7.6. *Эффективная реализация решета Эратосфена ($O(n)$) ***
 - 2.7.7. *Эффективная проверка числа на простоту ***
 - 2.7.8. *Быстрые алгоритмы разложения чисел на простые множители. Ро-эвристика ***
- 2.8. Алгоритмы на строках
 - 2.8.1. *Поиск подстроки в строке. Наивный метод **
 - 2.8.2. *Алгоритмы поиска подстроки в строке за $O(N+M)$ ***
 - 2.8.3. *Периодические и циклические строки ***
 - 2.8.4. *Алгоритм поиска нескольких подстрок за линейное время ***
- 2.9. Алгоритмы на графах
 - 2.9.1. *Вычисление длин кратчайших путей в дереве*
 - 2.9.2. *Обход графа в ширину и в глубину*
 - 2.9.3. *Способы реализации поиска в ширину (“наивный” и с очередью) **
 - 2.9.4. *Проверка графа на связность **

- 2.9.5. *Алгоритмы поиска кратчайшего пути во взвешенных графах **
- 2.9.6. *Топологическая сортировка графа, нахождение компонент сильной связности и построение диаграммы порядка ***
- 2.9.7. *Циклы отрицательной длины – критерий наличия, поиск ***
- 2.9.8. *Задача о синхронизации времени и задача о системе неравенств ***
- 2.9.9. *Алгоритм поиска эйлера цикла (в том числе лексикографически минимального) ***
- 2.9.10. *Нахождение транзитивного замыкания графа ***
- 2.9.11. *Алгоритмы нахождения взвешенных остовных деревьев ***
- 2.9.12. *Алгоритмы отыскания компонент двусвязности, точек сочленения, мостов с помощью поиска в глубину ***
- 2.9.13. *Алгоритм нахождения максимального паросочетания и минимального вершинного покрытия в двудольном графе ***
- 2.9.14. *Поиск максимального потока в сети ***

2.10. Динамическое программирование

- 2.10.1. *Основная идея динамического программирования. Рекурсивная реализация и разворачивание в цикл **
- 2.10.2. *Задачи с монотонным направлением движения в таблице **
- 2.10.3. *Задача о рюкзаке – решение методом динамического программирования **
- 2.10.4. *Оптимизация решения задачи динамического программирования на примере задачи о рюкзаке (исключение лишних параметров) ***
- 2.10.5. *Восстановление решения в задачах динамического программирования ***
- 2.10.6. *Общая схема решения задач динамического программирования ***

2.11. Алгоритмы теории игр

- 2.11.1. *Динамическое программирование и полный перебор как методы решения игровых задач ***
- 2.11.2. *Игры на ациклическом графе ***
- 2.11.3. *Оценка позиций. Альфа-бета отсечение ***

2.12. Геометрические алгоритмы

- 2.12.1. *Алгоритмы определения совпадения точек, лучей, прямых и отрезков*
- 2.12.2. *Представление точек, прямых и отрезков на плоскости **
- 2.12.3. *Нахождение расстояний между объектами на плоскости ***
- 2.12.4. *Алгоритмы определения пересечения отрезков на плоскости ***
- 2.12.5. *Алгоритмы вычисления площади многоугольника с заданными координатами вершин. Случай целочисленной решетки (формула Пика) ***

- 2.12.6. *Алгоритмы построения выпуклой оболочки (алгоритмы Грэхема и Джарвиса) ***
- 2.12.7. *Окружности на плоскости, пересечение их с другими геометрическими объектами ***
- 2.12.8. *Эффективный алгоритм нахождения пары ближайших точек на плоскости ***

3. Основы программирования

3.1. Языки программирования

- 3.1.1. *Классификация языков программирования*
- 3.1.2. *Процедурные языки*
- 3.1.3. *Основы синтаксиса и семантики языков высокого уровня **
- 3.1.4. *Формальные методы описания синтаксиса: форма Бэкуса-Наура ***
- 3.1.5. *Объектно-ориентированные языки ***

3.2. Основные конструкции программирования

- 3.2.1. *Переменные, типы, выражения и присваивания*
- 3.2.2. *Основы ввода/вывода*
- 3.2.3. *Операторы проверки условия и цикла*
- 3.2.4. *Функции и передача параметров **
- 3.2.5. *Структурная декомпозиция ***

3.3. Переменные и типы данных

- 3.3.1. *Концепция типа данных как множества значений и операций над ними*
- 3.3.2. *Свойства объявлений (связывание, область видимости, блоки и время жизни) **
- 3.3.3. *Обзор проверки типов **

3.4. Типы структур данных

- 3.4.1. *Примитивные типы*
- 3.4.2. *Массивы*
- 3.4.3. *Записи **
- 3.4.4. *Стратегии выбора подходящей структуры данных **
- 3.4.5. *Представление данных в памяти ***
- 3.4.6. *Статическое, автоматическое и динамическое выделение памяти ***
- 3.4.7. *Указатели и ссылки ***
- 3.4.8. *Связанные структуры ***
- 3.4.9. *Методы реализации стеков, очередей и хэш-таблиц ***
- 3.4.10. *Методы реализации графов и деревьев ***

3.5. Механизмы абстракции.

- 3.5.1. *Классы и объекты, замыкания **
- 3.5.2. *Процедуры, функции и итераторы как механизмы абстракции **
- 3.5.3. *Механизмы параметризации (ссылки и значения) **
- 3.5.4. *Модули в языках программирования **

3.6. Особенности программирования фундаментальных алгоритмов.

- 3.6.1. *Стратегии решения задач*
- 3.6.2. *Роль алгоритмов в процессе решения задач*
- 3.6.3. *Стратегии реализации алгоритмов **
- 3.6.4. *Реализация рекурсии **
- 3.6.5. *Стратегии отладки ***

4. Средства ИКТ

4.1. Цифровая логика

- 4.1.1. *Системы счисления*
- 4.1.2. *Компьютерная арифметика*
- 4.1.3. *Логические схемы **

4.2. Представление данных в памяти компьютера

- 4.2.1. *Биты, байты и слова **
- 4.2.2. *Представление числовых данных ***
- 4.2.3. *Системы с фиксированной и плавающей точкой ***
- 4.2.4. *Представление со знаковым битом и в дополнительном коде ***
- 4.2.5. *Представление нечисловых данных (коды символов, графические данные) ***
- 4.2.6. *Представление массивов и записей ***

4.3. Организация работы компьютера

- 4.3.1. *Принципы фон Неймана*
- 4.3.2. *Управляющее устройство: выборка инструкций, декодирование и выполнение **
- 4.3.3. *Набор инструкций и виды инструкций (манипуляция данными, управление, ввод-вывод) **
- 4.3.4. *Форматы инструкций ***
- 4.3.5. *Режимы адресации ***
- 4.3.6. *Механизм вызовов и возвратов из процедур ***
- 4.3.7. *Ввод-вывод и прерывания ***

4.4. Устройство памяти компьютера

- 4.4.1. *Организация основной памяти и операции с ней*
- 4.4.2. *Иерархия памяти **
- 4.4.3. *Кодирование данных, сжатие данных и целостность ***

4.4.4. *Кэш-память ***

4.5. Взаимодействие и коммуникации

4.5.1. *Интерфейс пользователя. Основы ввода-вывода информации. Основы скоростного клавиатурного ввода.*

4.5.2. *Введение в сетевые технологии*

4.5.3. *Внешняя память, физическая организация и устройства **

4.5.4. *Прямой доступ к памяти ***

5. Операционные системы

5.1. Основы операционных систем

5.1.1. *Роль и задачи операционных систем*

5.1.2. *Функционирование типичной операционной системы*

5.1.3. *Директории: содержимое и структура*

5.1.4. *Именование, поиск, доступ, резервное копирование **

5.2. Основные функции операционных систем

5.2.1. *Абстракции, процессы и ресурсы **

5.2.2. *Организация устройств **

5.2.3. *Защита, доступ и аутентификация **

5.3. Управление памятью

5.3.1. *Обзор физической памяти и аппаратного обеспечения, предназначенного для управления памятью **

5.3.2. *Страничная и сегментная организации памяти ***

5.3.3. *Кэширование ***

6. Основы технологии программирования

6.1. Программные средства и окружения

6.1.1. *Среды программирования*

6.1.2. *Инструментальные средства тестирования **

6.2. Проверка соответствия программного обеспечения

6.2.1. *Основы тестирования программ*

6.2.2. *Создание тестового плана и генерация тестов **

6.2.3. *Тестирование методом "черного ящика" и "белого ящика" **

6.2.4. *Тестирование элементов, интеграционное, системное тестирование и проверка соответствия ***

7. Методы вычислений и моделирование

7.1. Основы вычислительной математики

7.1.1. *Основные методы вычислительной математики*

- *вычисление периметра и площади плоских фигур*

- *вычисление объема плоских фигур **
- *вычисление значения и корней функции **
- 7.1.2. *Арифметика с плавающей точкой **
- 7.1.3. *Вычисление функций с шагом. Метод сеток ***
- 7.1.4. *Ошибка, устойчивость, сходимость ***
- 7.2. Введение в моделирование
 - 7.2.1. *Понятия модели и моделирования*
 - 7.2.2. *Основные типы моделей*
 - 7.2.3. *Компоненты компьютерной модели и способы их описания: входные и выходные переменные, переменные состояния, функции перехода и выхода, функция продвижения времени **
 - 7.2.4. *Основные этапы и особенности построения компьютерных моделей **
 - 7.2.5. *Основные этапы использования компьютерных моделей при решении практических задач **

8. Компьютерные сетевые технологии

- 8.1. Сети и телекоммуникации.
 - 8.1.1. *Сетевые устройства*
 - 8.1.2. *Среды передачи данных*
 - 8.1.3. *Использование паролей и механизмов контроля доступа*
 - 8.1.4. *Использование сетевых ресурсов*
 - 8.1.5. *Сетевые архитектуры **
 - 8.1.6. *Вопросы качества обслуживания: производительность, восстановление после сбоев ***
- 8.2. Беспроводные сети.
 - 8.2.1. *Специфические проблемы беспроводных и мобильных компьютеров*
 - 8.2.2. *Установка программ на мобильные и беспроводные компьютеры **
 - 8.2.3. *Беспроводные локальные сети и линии связи **

Олимпиадные задачи для школьного и муниципального этапов Олимпиады должны отличаться тематическим разнообразием и давать возможность использовать в процессе их решения знания и умения, характерные для основных этапов решения задач с помощью компьютеров. В частности, такими этапами являются:

- формализация задачи;
- выбор формального метода и разработка алгоритма решения задачи, включая оценку правильности и сложности алгоритма;
- программирование алгоритма и отладка программы;

- тестирование полученной программы.

Очевидно, что чем выше уровень Олимпиады, тем сложнее предлагаемые задачи и больший уровень знаний и умений требуется от участников. Но совершенно не правильно считать, что эта сложность возрастает только за счет программирования. Программирование здесь, как и в информатике в целом, играет важную, но не определяющую роль, и названный выше перечень знаний и умений участников в гораздо большей степени охватывает другие разделы информатики как науки.

6. Примеры олимпиадных задач

При разработке задач для школьного и муниципального этапов большую помощь могут оказать существующие печатные издания и имеющиеся в свободном доступе интернет-ресурсы, содержащие коллекции олимпиадных задач разного уровня сложности. Причем в качестве основы для разработки олимпиадной задачи могут использоваться даже задачи международных олимпиад. Дело в том, что сложность задач заключительных этапов и международных олимпиад в большинстве случаев определяется размерностью задачи. Уменьшив эту размерность, можно получить задачу, которая вполне под силу школьникам, которые только начинают свой путь в олимпиадной информатике.

Если говорить о печатных изданиях, содержащих в достаточном количестве олимпиадные задачи по информатике, то здесь можно порекомендовать книги, перечень которых представлен в списке рекомендуемой литературы. Среди них можно выделить книги издательства «Просвещение» (<http://prosv.ru/>), непосредственно посвященные всероссийской олимпиаде школьников по информатике [15 – 18] и книги, изданные издательством «БИНОМ. Лаборатория знаний» в рамках библиотечки олимпиадной информатики [6, 9, 11, 12, 21, 26 – 30, 33, 36] (<http://metodist.lbz.ru>).

Среди интернет-ресурсов полезными при разработке олимпиадных задач для школьного и муниципального этапа являются следующие сайты:

<http://algolist.manual.ru/olimp> (сайт «Олимпиадные задачи по программированию»);

<http://www.olympiads.ru/moscow> (сайт московских олимпиад по информатике);

<http://neerc.ifmo.ru/school> (сайт «Олимпиады по информатике. Санкт-Петербург, Россия»);

<http://contest.ur.ru> (сайт Уральских олимпиад по информатике);

<http://www.olympiads.ru> (сайт по олимпиадной информатике);

<http://www.olympiads.nnov.ru> (сайт «Олимпиадная информатика в Нижнем Новгороде»);

<http://acmp.ru> или <http://acm.dvpion.ru> (сайт «Школа программиста» для школьников Красноярского края);

<http://acmu.ru> (сайт «Олимпиады по информатике для школьников Ханты-Мансийского автономного округа»);

<http://olimpic.nsu.ru/nsu/archive/2005/index.shtml> (сайт открытой Всесибирской олимпиады по программированию им. И.В. Поттосина);

<http://imcs.dvgu.ru/works/school.html> (сайт школьных олимпиад, проводимых в Приморском крае);

<http://imcs.dvgu.ru/ru/event/jpa/2010/ai.html> (сайт ДВГУ для проведения соревнований по игровому ИИ для школьников);

<http://olymp.karelia.ru/pract.htm> (сайт школьных олимпиад Республики Карелия);

<http://school.sgu.ru> (сайт по алгоритмизации и программированию Саратовского государственного университета);

<http://www.olympiads.ru/moscow/2009/79/archive/index.shtml> (сайт с задачами московской олимпиады школьников по программированию для 7 – 9 классов).

Можно также воспользоваться сайтами, которые содержат не только коллекции олимпиадных задач, но и обеспечивают возможность проверки решений представленных там задач. К таким сайтам относятся:

<http://acm.timus.ru/> (сайт Уральского государственного университета, содержащий большой архив задач с различных соревнований по спортивному программированию);

<http://informatics.mccme.ru> (сайт дистанционной подготовки по информатике Московского института открытого образования и МЦНМО);

<http://imcs.dvgu.ru/cats> (сайт ДВГУ, содержащий архив задач с системой онлайн-проверки);

<http://acm.sgu.ru> (сайт Саратовского государственного университета, содержащий архив задач с системой онлайн-проверки).

7. Рекомендации по проверке и оцениванию решений задач

Методику проверки и систему оценивания решений задач школьного и муниципального этапов Олимпиады предоставляют организаторам и жюри этих этапов соответствующие предметно-методические комиссии. В случае автоматизированной проверки решений задач все необходимые для этого материалы должны поступить в распоряжение жюри как минимум за 7 дней до начала Олимпиады, чтобы члены жюри смогли настроить и проверить работоспособность соответствующего программного обеспечения.

7.1. Методика проверки решений задач участников

Методика проверки решений каждой олимпиадной задачи зависит от типа этой задачи. Если решением задачи является программа, то оценка правильности ее решения осуществляется путем исполнения программы с входными данными, соответствующими каждому тесту из представленного предметно-методической комиссией соответствующего

этапа комплекта тестов с последующим анализом получаемых в результате этого выходных файлов. Если решением задачи является набор выходных файлов для заданного в условии задачи набора входных файлов, то оцениваются только представленные на проверку выходные файлы.

Если участники Олимпиады должны сдавать на проверку решения в виде исходного текста программы на одном из разрешенных языков программирования, то проверка решений каждого участника должна осуществляться в следующей последовательности:

- компиляция исходного текста программы;
- последовательное исполнение программы с входными данными, соответствующими тестам из набора тестов для данной задачи, подготовленного предметно-методической комиссией по информатике соответствующего этапа;
- сравнение результатов исполнения программы на каждом тесте с правильным ответом.

При компиляции исходного текста программы, которую участник сдал на проверку, необходимо учитывать следующее.

1) Жюри должно использовать вполне определенные команды компиляции, соответствующие выбранному участником языку программирования. Таблица команд компиляции доводится до сведения всех участников перед началом каждого тура и должна содержаться в Памятке участнику.

2) Желательно учитывать, что размер файла с исходным текстом программы не должен превышать 256 Кбайт. Время компиляции программы не должно превышать одной минуты.

В случае нарушения принятых жюри ограничений решение участника считается неправильным и никакие баллы за эту задачу участнику не начисляются. Информация об этих ограничениях также должна быть размещена в Памятке участнику.

При исполнении программы на каждом тесте, в первую очередь, жюри должно определить, нарушаются ли присутствующие в условии этой задачи ограничения на время работы программы на отдельном тесте и размер доступной программе памяти в процессе ее исполнения. В случае нарушения имеющих место ограничений баллы за этот тест участнику не начисляются.

Если приведенные в условии задачи ограничения не нарушаются в процессе исполнения программы с входными данными, соответствующими конкретному тесту, то после завершения исполнения программы осуществляется проверка правильности полученного ответа. Эта проверка может осуществляться как путем сравнения полученных выходных данных с правильными ответами, так и с использованием предоставляемых

предметно-методической комиссией соответствующего этапа проверяющих программ, если для проверки решений участников предполагается использовать специализированную программную среду соревнований с возможностью проверки решений в автоматическом режиме.

Все представленные на проверку решения участников сначала должны проходить предварительное тестирование на тестах из примера или примеров, приведенных в условии задачи. Если на этих тестах решение участника выдает правильный ответ, то тогда это решение принимается жюри на окончательную проверку, которая после завершения соответствующего тура осуществляется на всех тестах из заданного набора тестов для этой задачи. В противном случае, решение участника считается неверным, и за него участнику не начисляются какие-либо баллы.

При проверке решений участников с использованием специализированной программной среды соревнований процесс предварительной проверки осуществляется в течение тура по мере отправки решений на сервер соревнований. В зависимости от возможностей проверяющей системы на окончательную проверку может приниматься либо последнее прошедшее предварительное тестирование решение одной и той же задачи, либо то, которое он должен указать. В любом случае, участник Олимпиады должен быть проинформирован до начала тура, каким образом будет определяться решение, принятое проверяющей системой для окончательной проверки. Эту информацию также следует разместить в Памятке участнику.

При проверке задач школьного этапа для обучающихся 5 – 6 классов, в процессе решения которых предполагается использование программных систем типа КуМир, Скретч, Роботландия и т.п., муниципальная комиссия должна предоставить жюри школьного этапа методику проверки таких задач с учетом установленных форм представления их решений участниками.

7.2. Система оценивания решений участников

Система оценивания решений каждой олимпиадной задачи школьного и муниципального этапов Олимпиады должна предоставляться жюри предметно-методической комиссией соответствующего этапа. Система оценивания той или иной задачи в значительной степени определяется ее типом и установленной формой представления результатов ее решения.

При разработке системы оценивания предметно-методическая комиссия по информатике соответствующего этапа сначала должна установить максимальный балл за полное решение задачи, а затем распределить его между различными вариантами частичных решений. При определении максимального количества баллов за задачу можно использовать

два подхода. Первый подход основан на предварительной оценке членами предметно-методической комиссии относительной сложности отобранных на туры задач и последующем назначении максимального количества баллов за задачу с учетом этих оценок. Второй подход заключается в том, что каждая задача оценивается одинаково, например из 100 баллов, независимо от того, какого мнения относительно их сложности имеют члены жюри.

В последнее время на региональном и заключительном этапах Олимпиады, а также на международных олимпиадах по информатике наиболее часто используется второй подход, то есть каждая задача оценивается из 100 баллов, независимо от ее предполагаемой сложности. Это объясняется следующими фактами.

Сказать перед началом тура, какая задача будет для участников сложной, а какая – нет, практически не возможно, за исключением очевидных случаев или когда уровень подготовленности участников Олимпиады известен. Попытки вводить различные коэффициенты сложности задачи до тура и после тура были на первых всесоюзных и всероссийских олимпиадах по информатике, но потом от этого отказались, так как на результаты участников влияют многие факторы, учесть которые введением коэффициентов сложности перед началом тура очень сложно. Более того, нередко были случаи, когда простая, по мнению жюри, задача оказывалась для всех участников достаточно сложной.

Нередки также случаи, когда при задании в явном виде уровня сложности задачи (максимальное количество баллов, которое может получить участник) многие неуверенные в своих силах участники начинают решать задачи, которые оценены меньшим количеством баллов, в то время как сильные участники – наоборот. В результате как те, так и другие, могут потратить много времени на решение первой выбранной ими задачи и не дойти до других задач не потому, что они сложные, а потому, что не хватило на них времени. К тому же, на олимпиадах по информатике разного уровня не так уж редки случаи, когда сильные участники самую простую задачу решали и не смогли решить. Но это уже проблемы психологической устойчивости участников, которые играют не менее важную роль, нежели уровень подготовленности к соревнованиям.

Распределение максимального количества баллов за задачу между различными вариантами частичных решений в общем случае базируется на системе тестов. Если результатом решения задачи является программа, то комплекты тестов разрабатываются таким образом, чтобы жюри соответствующего этапа без проблем могло в максимальной степени оценить все возможные типы алгоритмов, которые могут быть использованы в решениях участников и продифференцировать полученные участниками решения по степени их корректности и эффективности. В общем случае в комплекте тестов для каждой задачи выделяются следующие группы тестов:

- 1) тесты минимальной размерности (тривиальные тесты);
- 2) тесты на частные случаи, позволяющие выявить особенности используемых алгоритмов;
- 3) тесты на точность вещественных вычислений, если исходные данные таковы, что вызывают численную неустойчивость алгоритмов;
- 4) тесты, выявляющие особенности использования конкретных систем программирования при реализации алгоритмов решения задачи (например, неэффективная реализация потокового ввода-вывода и линейных контейнеров в C++);
- 5) общие тесты (достаточно случайные тесты, разные по размеру: от простых тестов до сложных);
- 6) тесты, проверяющие наличие эвристик в алгоритмах;
- 7) тесты максимальной размерности (тесты с использованием максимальных значений входных переменных, позволяющие оценить эффективность предложенных алгоритмов или их работоспособность при максимальной размерности задачи).

Распределение максимального количества баллов за задачу между всеми группами тестов и отдельными тестами внутри каждой группы представляется в виде таблицы, в которой каждому тесту и группе тестов ставится в соответствие определенное количество баллов. Такое распределение строится следующим образом: сначала максимальное количество баллов за задачу распределяется между всеми группами тестов, а затем между тестами внутри каждой группы.

При распределении максимального количества баллов за задачу между всеми группами тестов учитывается следующий принцип: правильное решение для всех ограничений из условия задачи должно набирать полный балл, в то время как правильное для определенной размерности входных данных, но неэффективное в целом решение задачи, должно набирать ориентировочно 30–70% баллов.

Поскольку каждый тест в группе используется для проверки вполне определенного свойства алгоритма решения задачи, то баллы внутри группы распределяются с учетом важности этого свойства для решения задачи в целом. В случае правильного ответа на тесты из конкретной группы или определенные тесты внутри этой группы участнику начисляется установленное для этой группы или теста количество баллов, в противном случае – нуль баллов.

Общая оценка за решение отдельной задачи конкретным участником складывается из суммы баллов, начисленных ему по результатам исполнения тестов из всех групп тестов для

этой задачи. Итоговая оценка проверки решений всех задач Олимпиады для каждого участника формируется как сумма полученных этим участником баллов за каждую задачу.

Итоговые результаты проверки решений всех задач заносятся в соответствующую тому или иному классу обучения участников итоговую таблицу, представляющую собой ранжированный список участников, расположенных по мере убывания набранных ими баллов. Участники с равным количеством баллов располагаются в алфавитном порядке и разделяют общее место.

7.3. Технология проверки решений участников

Существуют различные способы проверки решений участников. Если по условию задачи ее решением должна быть программа, то самый простой способ, но в то же время самый трудоемкий, заключается в последовательном запуске проверяемой программы на каждом тесте из заданного комплекта тестов для этой задачи. Для этого способа вполне достаточно иметь для каждого теста файл с входными данными и файл с соответствующими выходными данными. Если учесть, что для каждой задачи эти файлы предоставляются предметно-методической комиссией по информатике соответствующего этапа, то жюри при наличии достаточного количества членов вполне могут справиться с задачей проверки решений участников таким «ручным» способом.

Если по условию задачи ее решением является набор выходных файлов, то проверка сданного участником на проверку файла осуществляться путем его сравнения с правильным выходным файлом.

Конечно, описанный способ достаточно трудоемкий, но тот факт, что решения участников сначала проверяются на одном или двух тестах из условия задачи, и только в случае успешного прохождения этих тестов решение далее проверяется на всех тестах из заданного набора, в определенной степени уменьшает объем необходимой работы. Более продуктивным выходом из создавшегося положения является автоматизация процесса проверки решений участников. Как минимум, это можно сделать с помощью командных файлов, которые следует подготовить предметно-методическим комиссиям соответствующего этапа и включить в состав комплекта материалов для проверки решений участников членами жюри.

В настоящее время во многих субъектах РФ вопрос с автоматизированной проверкой решений участников успешно решается, и специализированные системы проведения соревнований используются достаточно широко. Организаторы школьного или муниципального этапов, которые только начинают осваивать современные информационные технологии при проведении олимпиад по информатике, могут решать эту проблему одним из следующих способов:

1) Разработать своими силами простейшую программную систему автоматической проверки решений олимпиадных задач по информатике, ориентируясь на материалы предметно-методической комиссии соответствующего этапа.

2) Использовать для проверки решений участников одну из свободно распространяемых программных систем проведения олимпиад по информатике, информацию о которых можно найти либо в интернете, либо обратившись в региональную предметно-методическую комиссию по информатике.

3) Закупить одну из имеющихся в стране коммерческих программных систем для проведения олимпиад по информатике.

Какой способ лучше – это выбор организаторов школьного или муниципального этапа, поскольку везде есть свои преимущества и недостатки. Центральная предметно-методическая комиссия готова оказать консультационную помощь в решении этой проблемы, если представители региональной предметно-методической комиссии к ней обратятся.

Поскольку в функции предметно-методических комиссий по информатике соответствующего этапа Олимпиады не входит обеспечение школьного и муниципального этапов программными системами проведения олимпиад по информатике, то единственное, чем они могут помочь организаторам школьного или муниципального этапов в решении вопроса автоматизации проверки решений участников – это предоставить жюри проверяющие программы для каждой задачи, которые могут быть либо интегрированы в большинство уже существующих в стране аналогичных систем, либо использованы в самостоятельно разработанных системах.

Следует заметить, что вопрос обеспечения школьного или муниципального этапов автоматизированными системами проверки решений участников не должен решаться только членами жюри соответствующего этапа накануне проведения Олимпиады. Организаторы соответствующего этапа должны предусмотреть решение этого вопроса задолго до проведения Олимпиады, поскольку для создания или приобретения и освоения такой системы требуется определенное время и дополнительные материальные и финансовые ресурсы, которых перед проведением соответствующего этапа у его организаторов может не оказаться. Здесь можно пойти по пути ряда субъектов РФ, которые на региональном уровне приобретают или разрабатывают типовую программную систему проведения Олимпиад по информатике, и затем тиражируют ее в учреждения образования, обеспечивая проведение школьного или муниципального этапа в своем регионе по единой технологии.

Возможны также и другие пути решения названной проблемы, например, и на муниципальном уровне. Однако в этом случае предметно-методическим комиссиям

соответствующих этапов при формировании комплектов тестов и проверяющих программ необходимо будет учитывать разнообразие используемых систем проверки решений участников в различных учреждениях образования, проводящих школьный или муниципальный этапы.

Что касается подготовки предметно-методическими комиссиями соответствующих этапов материалов для автоматизированной проверки решений участников, то центральная предметно-методическая комиссия по информатике рекомендует направлять их в адрес жюри на компакт-диске. Материалы для каждой задачи должны быть представлены в отдельном каталоге. Каталог каждой задачи может включать:

1) Каталог «preliminary», содержащий тесты из примеров в условии задачи, предназначенные для предварительной проверки решений участника во время тура. Каждый тест из примера должен содержаться в отдельном файле, входные файлы называются «01», «02» и т.д. Файлы с правильными ответами называются «01.a», «02.a» и т.д. Тесты пронумерованы в том же порядке, в котором они следуют в условии задачи.

2) Каталог «tests», содержащий тесты для окончательной проверки и оценивания решений участников, правильные ответы и программы-генераторы, использовавшиеся для создания тестов. Программы-генераторы приведены исключительно для ознакомления жюри и организаторов школьного или муниципального этапа с методами получения тестов. Их запуск и использование для регенерации тестов не требуется, так как комплект материалов уже содержит сгенерированные тесты и ответы к ним. Каждый тест должен содержаться в отдельном файле. Входные файлы называются «01», «02» и т.д. Файлы с правильными ответами называются «01.a», «02.a» и т.д. Программы генераторы содержатся в файлах, написанных на различных языках программирования, их компиляция и метод использования обычно описаны в командном файле, например, «doall.cmd», содержащемся в том же каталоге.

3) Исходный текст проверяющей программы для данной задачи «check.dpr».

4) Примеры правильных и неправильных решений. Каждое решение должно находиться в отдельном файле. Этот файл имеет, например, имя, построенное по маске «problem.ext», где «problem» – идентификатор задачи, обычно он совпадает с названием каталога, в котором находятся материалы задачи, «ext» – расширение файла, соответствующее языку программирования, на котором написано решение. Решения, которые являются неправильными или неоптимальными, также имеют суффикс, обычно указывающий на проблемы с этим решением, например «problem_wrong.ext» (неправильное решение), «problem_slow.ext» (медленное решение), «problem_bug.ext» (решение, содержащее некоторую типичную ошибку). Решения предоставляются для ознакомления членов жюри с возможной реализацией правильных и неправильных решений. Их

использование для генерирования правильных ответов на тесты не требуется, так как материалы содержат готовые правильные ответы на все тесты.

5) Каталог «reference», содержащий эталонное решение задачи, которое должно использоваться для подбора ограничения по времени в случае существенных различий в быстродействии проверяющего компьютера, используемого региональным жюри, и рекомендуемого.

Проверяющая программа для каждой задачи, как правило, представляет собой файл, написанный на определенном языке программирования. Для компиляции проверяющей программы следует использовать соответствующий компилятор. В своей работе проверяющие программы могут использовать подключаемые модули, которые также должны входить в состав материалов, распространяемых предметно-методическими комиссиями соответствующего этапа.

Запуск проверяющей программы может осуществляться следующим образом. В пустой каталог необходимо скопировать проверяющую программу (check.exe), входной файл, ответ для которого следует проверить (input), выходной файл, созданный программой участника (output) и файл с правильным ответом (answer). После этого проверяющая программа запускается с четырьмя параметрами: «check.exe input output answer result». Программа завершает свою работу с одним из трех возможных кодов возврата:

- 0 – ответ участника является правильным;
- 1 – ответ участника удовлетворяет формату вывода, но является неправильным, в этом случае участнику отправляется сообщение «Неверный ответ»;
- 2 – ответ участника не удовлетворяет формату вывода, в этом случае участнику отправляется сообщение «Ошибка формата выходных данных»;
- 3 – найдена внутренняя ошибка.

Файл «result» после завершения программы будет содержать сформированный проверяющей программой комментарий о причинах, по которым был выдан соответствующий отклик. Этот комментарий не должен сообщаться участникам и служит только для справки членам жюри.

Проверяющая программа check.exe используется как на этапе предварительной проверки, так и на этапе окончательной проверки и оценки решений участников. Отличие заключается в том, что при окончательной проверке при возникновении любой ошибки вместо формирования участнику сообщения, система просто помечает тест как не пройденный, и по результатам проверки участнику начисляются баллы за успешно пройденные тесты в соответствии с системой оценивания, разработанной предметно-методической комиссией соответствующего этапа.

Для задач с открытым входом, то есть, когда решением задачи является набор выходных файлов, проверяющая программа для предварительной и окончательной проверки могут несколько отличаться друг от друга. Для предварительной проверки может использоваться проверяющая программа, например, с именем «checkPreliminary.dpr», работа которой осуществляется следующим образом. В пустой каталог копируются исполняемый файл проверяющей программы (checkPreliminary.exe), входной файл соответствующего теста (input), выходной файл, присланный участником на проверку (output) и файл с правильным ответом (answer). Программа запускается с четырьмя параметрами: «checkPreliminary input output answer result». Программа завершает свою работу с одним из двух возможных кодов возврата:

- 0 – ответ участника удовлетворяет формату вывода;
- 2 – ответ участника не удовлетворяет формату вывода, в этом случае участнику отправляется сообщение «Ошибка формата выходных данных».

Файл «result» будет содержать сформированный проверяющей программой комментарий о причинах, по которым был выдан соответствующий отклик. Этот комментарий не должен сообщаться участникам и служит только для справки членам жюри и оргкомитета соответствующего этапа Олимпиады. Если файл удовлетворяет формату вывода, то он принимается на проверку.

После окончания тура осуществляется окончательная проверка выходного файла. Для этого может использоваться, например, основная проверяющая программа с именем «check.dpr», работа которой осуществляется следующим образом. В пустой каталог копируются исполняемый файл проверяющей программы («check.exe»), входной файл, ответ для которого следует проверить (input), выходной файл, созданный программой участника (output), и файл с правильным ответом (answer). После этого проверяющая программа запускается с четырьмя параметрами: «check.exe input output answer result». Программа завершает свою работу с одним из двух возможных кодов возврата:

- 0 – ответ участника является правильным;
- 1 – ответ участника удовлетворяет формату вывода, но является неправильным.

Файл «result» после завершения программы будет содержать сформированный проверяющей программой комментарий о причинах, по которым был выдан соответствующий отклик. По результатам проверки участнику начисляются баллы за успешно пройденные тесты в соответствии с системой оценивания, разработанной предметно-методической комиссией соответствующего этапа.

Опыт использования в различных субъектах РФ систем автоматической проверки решений участников показал, что по своим функциональным возможностям и вариантам реализации такие системы могут отличаться друг от друга, но все они настроены на

использование проверяющих программ, о которых шла речь выше. Более того, можно выделить основные функции таких систем, которые характерны для многих из них. В частности, в процессе предварительной проверки решений участников, представленных в виде программ, такие системы должны последовательно выполнять следующие действия:

1) Скомпилировать программу участника, используя приведенную в Памятке участнику команду для соответствующего языка программирования. Если компиляция программы участника завершается неудачно, участнику сообщается результат «Ошибка компиляции». Возможно предоставление участнику вывода компилятора в стандартный поток вывода и стандартный поток ошибок. Если компиляция завершилась успешно, программа проверяется на тестах из примера.

2) Осуществить последовательную проверку программы участника на всех тестах из примера. Проверка на одном тесте осуществляется следующим образом. В пустой каталог копируется исполняемый файл программы участника и тестовый входной файл. Тестовый файл должен иметь имя, указанное в условии задачи. Далее программа участника запускается, и проверяющая система отслеживает соблюдение программой существующих ограничений, связанных с запретом на создание каталогов и временных файлов при работе программы, а также любое использование сетевых средств и выполнение других действий, нарушающих работу самой проверяющей системы.

3) Обеспечить контроль времени работы программы участника и объема используемой памяти. Если время работы программы превысило ограничение, указанное в условии задачи, выполнение программы участника прерывается и участнику отправляется сообщение «Превышено время работы». Если количество используемой памяти превысило ограничение, указанное в условии задачи, то выполнение программы участника также прерывается и участнику отправляется сообщение «Превышен максимальный объем используемой памяти».

4) Проверить, создала ли программа участника и самостоятельно обработала исключительную ситуацию (exception). Если программа участника создала и самостоятельно не обработала исключительную ситуацию (exception), выполнение программы участника прерывается и участнику отправляется сообщение «Ошибка времени исполнения».

5) Проверить, завершила ли программа участника работу с нулевым кодом возврата. Если программа участника завершила работу с ненулевым кодом возврата, участнику отправляется сообщение «Ошибка времени исполнения».

6) Проверить, создала ли программа участника в каталоге, в котором она была запущена, выходной файл с именем, указанным в условии задачи, если программа участника завершила работу за отведенный период времени, не превысила максимальный объем памяти и завершила работу с нулевым кодом возврата. Если файл с указанным именем не найден,

участнику отправляется сообщение «Ошибка формата выходных данных». Если выходной файл создан, то осуществляется проверка его корректности. Для этого используется соответствующая проверяющая программа.

7) Сообщить участнику о результатах проверки его программы. Если программа участника выдает правильный ответ на всех тестах из примера, то она может быть принята на окончательную проверку. В этом случае участнику отправляется сообщение «Принято на проверку», а тестирующая система запоминает решение участника как последнее принятое решение по данной задаче. В противном случае участнику отправляется сообщение в соответствии с описанными выше правилами. При этом участнику помимо типа ошибки сообщается номер теста из примера, на котором произошла ошибка.

При окончательной проверке решений участников, представленных в виде программ, которая осуществляется после тура, программная система проведения олимпиад по информатике должна проверить на основных тестах последнее принятое на проверку решение участника по каждой задаче. Выполняемые системой функции в этом случае во многом повторяют вышеописанные. Однако при возникновении любой ошибки вместо отправки участнику сообщения, система просто помечает тест как не пройденный. Кроме того, по результатам окончательной проверки система начисляет участнику баллы за успешно пройденные тесты.

Сказанное выше можно распространить и на проверку решений участников, представленных в виде набора выходных файлов. Однако из описанных выше функций программных систем для проведения Олимпиад по информатике остаются только две последние с небольшими изменениями. В частности, в процессе предварительной проверки осуществляется только контроль формата присланного участником выходного файла. Если этот файл удовлетворяет формату вывода, то он принимается на окончательную проверку. В противном случае, по результатам предварительной проверки участнику отправляется сообщение «Ошибка формата выходных данных».

Если в процессе подготовки и использования во время соревнований системы автоматической проверки решений задач у членов жюри возникают вопросы к комплектам тестов и проверяющим программам, то они должны быть сразу адресованы соответствующей предметно-методической комиссии любым доступным способом. Предметно-методическая комиссия должна в кратчайшие сроки рассмотреть поступившие в ее адрес вопросы и дать окончательное решение по ним. Не допускается внесение каких-либо изменений в систему оценивания без согласования с предметно-методической комиссией соответствующего этапа.

В заключение хотелось бы отметить, что если нет возможности использовать при проведении школьного или муниципального этапа какой-либо готовой программной

системы проведения Олимпиады по информатике, то разработка простейшей системы, позволяющей осуществлять окончательную проверку решений участников в автоматическом режиме, является не такой уж сложной задачей и по силам любой муниципальной или региональной предметно-методической комиссии по информатике с привлечением специалистов соответствующей квалификации. Наличие такой системы в субъекте РФ позволит на должном уровне осуществлять проверку решений участников не только на школьном и муниципальном этапах, но и региональном этапе Олимпиады по информатике.

Список рекомендуемой литературы

1. Алексеев А.В., Беляев С.Н. Подготовка школьников к олимпиадам по информатике с использованием веб-сайта: учебно-методическое пособие для учащихся 7-11 классов. – Ханты-Мансийск: РИО ИРО, 2008. – 284 с.
2. Андреева Е.В., Босова Л.Л., Фалина И.Н. Математические основы информатики. Элективный курс: Учебное пособие. – М.: БИНОМ. Лаборатория Знаний, 2007. – 312 с.
3. Арсак Ж. Программирование игр и головоломок. – М.: Наука, 1990. – 224 с.
4. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. — Пер. с англ. — М.: Мир, 1979. — 536 с.
5. Бентли Д. Жемчужины творчества программистов: пер. с англ. – М.: Радио и связь, 1990. – 224 с.
6. Босова Л.Л., Босова А.Ю., Коломенская Ю.Г. Занимательные задачи по информатике. – М.: БИНОМ. Лаборатория знаний. 2007. – 119 с.
7. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию/ Под ред. акад. Б.Н. Наумова.- 2-е изд., доп. и пераб. – М.: Наука, гл. ред. физ.-мат. лит., 1990. – 208 с.
8. Великович Л.С., Цветкова М.С. Программирование для начинающих. – М.: БИНОМ. Лаборатория знаний. 2007. – 287 с.
9. Волчёнков С.Г., Корнилов П.А., Белов Ю.А. и др. Ярославские олимпиады по информатике. Сборник задач с решениями. – М.: БИНОМ. Лаборатория знаний. 2010. – 405 с.
10. Долинский М.С. Алгоритмизация и программирование на Turbo Pascal: от простых до олимпиадных задач: Учебное пособие. – СПб.: Питер Принт, 2004. – 240 с.
11. Задачи по программированию /С.М. Окулов, Т.В. Ашихмина, Н.А. Бушмелева и др.; Под ред. С.М. Окулова. – М.: БИНОМ. Лаборатория знаний, 2006. – 820 с.
12. Златопольский Д. М. Программирование: типовые задачи, алгоритмы, методы. – М.: БИНОМ. Лаборатория знаний, 2007. – 223 с.
13. Иванов С.Ю., Кирюхин В.М., Окулов С. М. Методика анализа сложных задач по информатике: от простого к сложному // Информатика и образование. 2006. №10. С. 21 – 32.
14. Кирюхин В.М. Всероссийская олимпиада школьников по информатике. М.: АПК и ППРО, 2005. –212 с.
15. Кирюхин В.М. Информатика. Всероссийские олимпиады. Выпуск 1. – М.: Просвещение, 2008. – 220 с. – (Пять колец).

16. Кирюхин В.М. Информатика. Всероссийские олимпиады. Выпуск 2. – М.: Просвещение, 2009. – 222 с. – (Пять колец).
17. Кирюхин В.М. Информатика. Всероссийские олимпиады. Выпуск 3. – М.: Просвещение, 2010. – 201 с. – (Пять колец). (Планируется к выпуску в конце 2010 года).
18. Кирюхин В.М. Информатика. Международные олимпиады. Выпуск 1. – М.: Просвещение, 2009. – 239 с. – (Пять колец).
19. Кирюхин В.М., Окулов С. М. Методика анализа сложных задач по информатике // Информатика и образование. 2006. №4. С. 42 – 54.
20. Кирюхин В.М., Окулов С. М. Методика анализа сложных задач по информатике // Информатика и образование. 2006. №5. С. 29 – 41.
21. Кирюхин В.М., Окулов С. М. Методика решения задач по информатике. Международные олимпиады. – М.: БИНОМ. Лаборатория знаний, 2007. – 600 с.
22. Кирюхин В.М., Цветкова М.С. Всероссийская олимпиада школьников по информатике в 2006 году. – М.: АПК и ППРО, 2006. – 152 с.
23. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999. – 960с.
24. Меньшиков Ф.В. Олимпиадные задачи по программированию. – СПб.: Питер, 2006. – 315 с.
25. Московские олимпиады по информатике. 2002 – 2009. / Под ред. Е.В. Андреевой, В.М. Гуровица и В.А. Матюхина. – М.: МЦНМО, 2009. – 414 с.
26. Окулов С.М. Основы программирования. – М.: БИНОМ. Лаборатория знаний, 2005. – 440 с.
27. Окулов С.М. Программирование в алгоритмах. – М.: БИНОМ. Лаборатория знаний. 2002. – 341 с.
28. Окулов С.М. Дискретная математика. Теория и практика решения задач по информатике: учебное пособие. – М.: БИНОМ. Лаборатория знаний. 2008. – 422 с.
29. Окулов С.М. Алгоритмы обработки строк: учебное пособие. – М.: БИНОМ. Лаборатория знаний, 2009. – 255 с.
30. Окулов С.М., Лялин А.В. Ханойские башни. – М.: БИНОМ. Лаборатория знаний. 2008. – 245 с. (Развитие интеллекта школьников).
31. Пинаев В.Н. Олимпиадные задачи по программированию: Учебное пособие / РГАТА. – Рыбинск, 1997. – 41 с.
32. Просветов Г.И. Дискретная математика: задачи и решения: учебное пособие. – М.: БИНОМ. Лаборатория знаний. 2008. – 222 с.

33. Пупышев В.В. 128 задач по началам программирования. – М.: БИНОМ. Лаборатория знаний. 2009. – 167 с.
34. Рейнгольд Э. Комбинаторные алгоритмы: теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део. – М.: Мир, 1980. – 476 с.
35. Скиена С.С., Ревилла М.А. Олимпиадные задачи по программированию. Руководство по подготовке к соревнованиям. – М.: Кудиц-образ, 2005. – 416 с.
36. Столяр С.Е., Владыкин А.А.. Информатика. Представление данных и алгоритмы. – СПб.: Невский Диалект; М.: БИНОМ. Лаборатория знаний. 2007. – 382 с.
37. Сулейманов Р.Р. Организация внеклассной работы в школьном клубе программистов: методическое пособие. – М.: БИНОМ. Лаборатория знаний. 2010. – 255 с.
38. Уззерелл Ч. Этюды для программистов. – М.: Мир, 1982. – 288 с.
39. Шень А. Программирование: теоремы и задачи. – М.: МЦНМО, 1995. – 264 с.