

ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ EJUDGE

Л. В. Горчаков, А. Н. Стась, Д. В. Карташов

Томский государственный педагогический университет, Томск

Рассматривается актуальная проблема обучения программированию студентов. Одним из условий успешного обучения является формирование алгоритмического мышления. Одним из необходимых условий его формирования является самостоятельное решение задач разной степени сложности, поэтому при обучении программированию важно особое внимание уделить организации самостоятельной работы. Предлагается подход, основанный на использовании специализированной системы Ejudge, позволяющей в автоматическом режиме осуществлять проверку правильности работы учебных программ. Навыки работы с такой системой также необходимы для участия в олимпиадах по программированию.

Ключевые слова: *обучение программированию, алгоритмическое мышление, тестирующие системы, оценочные средства, олимпиадные задачи.*

В данный период развития общества информатизацией пропитаны все сферы человеческой деятельности. Очевидно, что для нормальной социальной адаптации выпускников в любой социальной сфере требуется не только традиционная грамотность, но и развитая информационная культура.

Традиционно принято считать, что именно информационная культура является основным показателем знаний обучающихся в школьном курсе информатики. Но одной информационной культуры недостаточно даже для получения минимальных навыков в области программирования. В то же время именно технология программирования, по сути, является тем базисом, на основе которого реализуются все остальные информационные технологии. В основе программирования лежит понятие алгоритмизации, рассматриваемой в широком смысле как процесс разработки и описания алгоритма средствами заданного языка [1]. Поэтому одной из задач обучения информатике является формирование алгоритмической культуры. А алгоритмическая культура, в свою очередь, требует сформированности мышления особого типа – алгоритмического мышления [2, 3]. Центральным способом формирования алгоритмического мышления является решение специально подобранных задач, в том числе и в режиме самостоятельной работы [4–8].

Существенную помощь в организации самостоятельной работы в рамках обучения программированию может оказать использование специальных программных средств, позволяющих в автоматическом режиме проверять корректность разработанной программы. Одним из таких средств является система Ejudge [9]. Данная система является сетевой, разворачивается на сервере, работа с системой как со стороны администратора, так и со стороны пользователей, в роли которых выступают обучающиеся и преподаватели, осуществляется через веб-

интерфейсы. Такая технология обеспечивает дополнительные возможности в организации обучения. С одной стороны, обучаемый может работать с системой из любой точки, в которой обеспечен доступ к сети Интернет. В то же время все действия обучаемого фиксируются на сервере и доступны для преподавателя.

Метод проверки правильности представленной программы в системе достаточно простой. Создается система тестов, каждый из которых содержит входные данные и выходные данные, которые должны быть выданы правильно работающей программой. Обучаемый через веб-интерфейс прикрепляет файл с текстом программы. Программа компилируется на сервере и запускается на выполнение с входными данными, представленными в каждом из тестов. Выходные данные сравниваются с эталоном с помощью специальной подпрограммы, называемой чекером. В системе реализованы компиляторы для подавляющего большинства популярных в настоящее время языков программирования – Pascal, Object Pascal, C, C++, Java, Python, Visual Basic, PHP и т. д. Имеется также множество готовых чекеров, позволяющих выбрать оптимальный способ сверки фактических выходных данных с эталоном, так, например, отдельные чекеры используются для сравнения целых и вещественных чисел, строк, последовательностей различных видов. Система является открытой, т. е. существует возможность подключения при необходимости новых компиляторов, добавления чекеров и т. д. Следует обратить внимание также на возможность задавать ограничения на время работы программы и на предельный объем памяти при прохождении одного теста. Это позволяет реализовать важнейшее требование к выполненному заданию – не просто решить задачу любым способом, а сделать это с использованием приемлемого по временной и (или) емкостной эффективности алгоритма.

Считается, что основное назначение данной системы и ее аналогов – это техническая поддержка проведения олимпиад (турниров) по программированию всех уровней. Действительно, система при грамотном использовании позволяет обеспечить возможность объективной и беспристрастной проверки выполнения задач участниками олимпиады. В системе поддерживается возможность начисления очков в соответствии с различными правилами проведения турниров по программированию, в том числе с использованием популярных регламентов Kirov и ACM.

Основная особенность регламента Kirov состоит в том, что определенные баллы начисляются за каждый пройденный тест независимо от результата прохождения остальных тестов. Такой подход фактически позволяет учитывать неполные решения. Регламент, на взгляд авторов, оптимален для применения на массовых соревнованиях школьников и студентов. Получение бонусов даже за неполные решения положительно сказывается на мотивации. Регламент ACM подразумевает, что задача считается решенной лишь в случае прохождения всех тестов. Этот подход оптимален для турниров среди профессионалов.

В случае использования системы для организации самостоятельной работы при обучении программированию турнир содержит последовательность учебных задач, которые должны быть выполнены студентом или школьником. При этом важно грамотно составить проверочные тесты.

Приведем примеры задач, загруженных в систему Ejudge.

Пример 1 (заимствован из демонстрационного варианта единого государственного экзамена по информатике 2016 г. [10]).

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передается положительное целое число – текущее показание прибора «Сигма». Необходимо вычислить бета-значение серии показаний прибора – минимальное четное произведение двух показаний, между моментами которых прошло не менее 6 минут. Если таких значений нет, то бета-значение равно нулю.

Формат входных данных:

Первая строка – N ($6 < N \leq 100000$) – количество показаний прибора.

Вторая строка – N чисел $M_1 \dots M_N$ ($0 < M_i \leq 1000$).

Формат выходных данных:

Одно число – бета-значение.

Пример входных и выходных данных

Входные данные	Выходные данные
8	8
1 2 3 4 5 6 7 8	

Простой алгоритм решения данной задачи основан на том, что значения помещаются в массив, и далее в массиве в двойном цикле (один цикл вложен в другой цикл) просматриваются все пары значений, находящиеся на расстоянии не менее шести. Отдельно следует рассмотреть случай, когда ни одно четное произведение не найдено. В этом случае результат полагаем равным нулю. Недостатки данного решения заключаются в том, что тратится память на хранение массива и что использование двойного вложенного цикла неэффективно. Таким образом, проверочные тесты должны быть составлены так, чтобы часть из них не удовлетворяла дополнительным ограничениям по времени и по памяти.

Эффективное решение основано на запоминании шести абсолютных максимумов просмотренных значений и шести максимумов среди четных значений без учета одного, двух, трех, четырех, пяти и шести последних значений соответственно. Максимальное четное произведение для текущего элемента всегда вычислимо как произведение текущего четного элемента на абсолютный максимум из предыдущих без учета шести последних элементов или как произведение текущего нечетного элемента на максимум среди четных без учета шести последних элементов. Данный способ решения эффективен как по времени, так и по емкости.

Пример 2. Дано игровое поле размерности $N \times M$ (N строк, M столбцов). В каждой клетке записано целое неотрицательное число. Фишка из текущей клетки может двигаться либо на одну клетку вниз, либо на одну клетку вправо. Требуется найти минимальный по сумме чисел в проходимых клетках путь из ячейки (1, 1) в ячейку (N , M).

Формат входных данных:

Первая строка – N ($1 \leq N \leq 100$) – количество строк.

Вторая строка – M ($1 \leq M \leq 100$) – количество столбцов.

N строк, вида:

$A_{11} \dots A_{1m}$

.....

$A_{n1} \dots A_{nm}$,

где $0 \leq A_{ij} < 30000$.

Формат выходных данных:

Одно число – длина минимального пути.

Примеры

Входные данные	Выходные данные
3	15
4	
1 2 4 3	
5 4 3 4	
2 1 4 3	

Данная задача также может быть решена несколькими способами. Неэффективный подход заключается в применении рекурсивного алгоритма по типу бэктрекинга (поиска с возвратом). Часть проверочных тестов должна быть подобрана таким образом, чтобы такое решение не удовлетворяло ограничениям по времени. Эффективное решение основано на применении идей динамического программирования. Минимальный путь до ячейки равен сумме числа в данной ячейке и минимума между длиной оптимального пути до ячейки, расположенной сверху, и ячейки, расположенной слева.

Отдельное преимущество дает использование Ejudge в процессе подготовки школьников и студентов к участию в олимпиадах по информатике. На практике основная часть серьезных предметных олимпиад по информатике фактически являются олимпиадами по программированию. И, как правило, олимпиадные задания проверяются с помощью современных программных средств, т. е. с помощью Ejudge или ее аналогов. Использование системы на всех этапах обучения программированию, а также непосредственно на подготовительных занятиях позволит будущим участникам олимпиады привыкнуть к особенностям таких систем. Так, необходимо привыкнуть в точности соблю-

дать требуемые в условиях задач форматы входных и выходных данных. Необходимо также учитывать правила проведения турнира. Так, если турнир проводится по регламенту Kirov, то целесообразно отсылать на проверку решения для отдельных частных случаев. Напротив, если турнир проводится по регламенту ACM, то решение следует отсылать только после его всесторонней проверки для всех случаев.

С 2015 г. система Ejudge внедрена и используется в обучении информатике студентов физико-математического факультета Томского государственного педагогического университета. Система успешно применяется для организации самостоятельной работы по дисциплинам «Программирование», «Технологии программирования», «Алгоритмы и структуры данных», «Теоретические основы прикладной математики и информатики». Опыт использования системы показывает, что ее использование способствует мотивации обучаемых к решению задач по программированию. Снижается нагрузка на преподавателя по проверке контрольных работ. Также система была успешно использована в процессе проведения предметных конкурсов в рамках олимпиад по информатике среди педагогических вузов Сибири в 2015 и 2016 гг.

Список литературы

1. Якименко О. В., Стась А. Н. Применение обучающих программ-тренажеров в обучении программированию // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2009. Вып. 1 (79). С. 54–56.
2. Стась А. Н., Долганова Н. Ф. Развитие алгоритмического мышления в процессе обучения будущих учителей информатики // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2012. Вып. 7 (122). С. 241–244.
3. Стась А. Н., Прусских О. Н. Формирование алгоритмического мышления в процессе обучения теории графов // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2012. Вып. 2 (117). С. 166–169.
4. Долганова Н. Ф., Стась А. Н. Основные дидактические принципы построения дисциплины «Элементы вычислительной геометрии» в условиях педагогического вуза // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2009. Вып. 1 (79). С. 56–58.
5. Стась А. Н., Долганова Н. Ф. О проблемах преподавания вычислительной геометрии в условиях педагогического вуза // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2007. Вып. 6 (69). С. 112–115.
6. Карташов Д. В., Стась А. Н. Методика обучения алгоритмам и структурам данных // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2015. Вып. 8 (161). С. 131–134.
7. Стась А. Н. Методика обучения разработки трансляторов // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2015. Вып. 8 (161). С. 76–81.
8. Стась А. Н., Горобец Л. С. Современные методы обучения вычислительной геометрии // Вестн. Томского гос. пед. ун-та (TSPU Bulletin). 2012. Вып. 2 (117). С. 170–174.
9. Система. URL: https://ejudge.ru/wiki/index.php/Система_ejudge (дата обращения: 20.12.2016).
10. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2016 года по информатике и ИКТ. URL: http://www.ege.edu.ru/common/upload/docs/2016/inf_11_2016.zip (дата обращения: 20.12.2016).

Горчаков Леонид Всеволодович, профессор кафедры информатики, доктор физико-математических наук, профессор, Томский государственный педагогический университет (ул. Киевская, 60, Томск, Россия, 634061).
E-mail: Gorchakov@phys.tsu.ru

Стась Андрей Николаевич, заведующий кафедрой информатики, кандидат тех. наук, ведущий инженер-программист лаборатории автоматизации управления и компьютеризации, Томский государственный педагогический университет (ул. Киевская, 60, Томск, Россия, 634061). E-mail: stasandr@tspu.edu.ru

Карташов Денис Васильевич, аспирант, Томский государственный педагогический университет (ул. Киевская, 60, Томск, Россия, 634061). E-mail: DeKar@tspu.edu.ru

Материал поступил в редакцию 24.05.2017.

DOI 10.23951/1609-624X-2017-9-109-112

PROGRAMMING TRAINING USING THE EJUDGE SYSTEM

L. V. Gorchakov, A. N. Stas, D. V. Kartashov

Tomsk State Pedagogical University, Tomsk, Russian Federation

The article focuses on the actual problem of programming training for students. A prerequisite for successful learning is the formation of algorithmic thinking. One of the necessary conditions for its formation is the independent solution of tasks of varying degrees of complexity, so during programming training it is important to pay special attention to the organization of independent study. Essential assistance in the organization of self-guided work within the programming training can be provided by the use of special software tools that allow to check the correctness of the developed program automatically.

We propose an approach based on the use of the specialized system 'Ejudge', which allows to check the correctness of the training program. This system is networked, deployed on the server, work with the system, which is implemented via web interfaces, can be conducted both by the administrator and by users, in the role of which students and teachers act. This technology provides additional opportunities in the organization of training. On the one hand, the trainee can work with the system from any place in which access to the Internet is provided. At the same time, all actions of the trainee are recorded on the server, and are accessible to the teacher. The skills of working with such a system are also necessary for participation in academic competition on programming.

Key words: *programming training, algorithmic thinking, testing systems, evaluation tools, olympiad problems*

References

1. Yakimenko O. V., Stas A. N. Primeneniye obuchayushchikh programm-trenazherov v obuchenii programmirovaniyu [Use of Computer Tutors in Teaching Programming]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2009, vol. 1 (79), pp. 54–56 (in Russian).
2. Stas A. N., Dolganova N. F. Razvitiye algoritmicheskogo myshleniya v protsesse obucheniya budushchikh uchiteley informatiki [Algorithmic thinking development when training computer science teachers]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2012, vol. 7 (122), pp. 241–244 (in Russian).
3. Stas A. N., Pruskikh O. N. Formirovaniye algoritmicheskogo myshleniya v protsesse obucheniya teorii grafov [Shaping the algorithmic thinking in the process of the education graph theory]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2012, vol. 2 (117), pp. 166–169 (in Russian).
4. Dolganova N. F., Stas A. N. Osnovnye didakticheskiye printsipy postroeniya distsipliny "Elementy vychislitel'noy geometrii" v usloviyakh pedagogicheskogo vuza [Basic Didactic Principles of Designing the Elements of Computational Geometry Course in Pedagogical Institute of Higher Education]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2009, vol. 1 (79), pp. 56–58 (in Russian).
5. Stas A. N., Dolganova N. F. O problemakh prepodavaniya vychislitel'noy geometrii v usloviyakh pedagogicheskogo vuza [The problems of computational geometry in the conditions of a pedagogical university]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2007, vol. 6 (69), pp. 112–115 (in Russian).
6. Kartashov D. V., Stas A. N. Metodika obucheniya algoritmam i strukturam dannykh [Methods of teaching algorithms and data structures] *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2015, vol. 8 (161), pp. 131–134 (in Russian).
7. Stas A. N. Metodika obucheniya razrabotke translyatorov [Methods of teaching the designing of translators]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2015, vol. 8 (161), pp. 76–81 (in Russian).
8. Stas A. N., Gorobets L. S. Sovremennyye metody obucheniya vychislitel'noy geometrii [Contemporary methods of teaching computer geometry]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2012, vol. 2 (117), pp. 170–174 (in Russian).
9. Sistema [System Ejudge]. URL: https://ejudge.ru/wiki/index.php/System_ejudge (accessed 20 December 2016).
10. Demonstratsionnyy variant kontrol'nykh izmeritel'nykh materialov edinogo gosudarstvennogo ekzamena 2016 goda po informatike i IKT [Demonstration version of control measuring materials of the unified state exam in 2016 on computer science and ICT]. URL: http://www.ege.edu.ru/common/upload/docs/2016/inf_11_2016.zip (accessed 20 December 2016).

Gorchakov L. V., Tomsk State Pedagogical University (ul. Kievskaya, 60, Tomsk, Russian Federation, 634061).
E-mail: Gorchakov@phys.tsu.ru

Stas A. N., Tomsk State Pedagogical University (ul. Kievskaya, 60, Tomsk, Russian Federation, 634061).
E-mail: Stasandr@tspu.edu.ru

Kartashov D. V., Tomsk State Pedagogical University (ul. Kievskaya, 60, Tomsk, Russian Federation, 634061).
E-mail: DeKar@tspu.edu.ru