

МЕТОДИКА ОБУЧЕНИЯ АЛГОРИТМАМ И СТРУКТУРАМ ДАННЫХ

Рассмотрена методика обучения студентов основным, часто используемым алгоритмам в процессе практического решения задач на ЭВМ и привития навыков эффективного программирования. Данная методика способствует развитию навыков алгоритмического мышления. Особенностью методики является последовательное изучение основных алгоритмов (сортировка и поиск, элементы теории информации и криптографии, рекурсивные алгоритмы и алгоритмы на графах) без привязки к конкретному языку программирования и последующая реализация пройденных алгоритмов на практике. Данный курс содержит лабораторный практикум, содержащий задания возрастающей трудности. Простые задания требуют от студента заполнения пропусков. Более сложные задания требуют навыков самостоятельной разработки и отладки программ.

Ключевые слова: методика обучения, алгоритмическое мышление, алгоритмы и структуры данных, сортировка и поиск, рекурсивные алгоритмы, алгоритмы на графах.

Исторически важнейшим понятием информатики и ИКТ является понятие алгоритма. Поэтому обучение алгоритмизации и программированию, а также способам их компьютерной реализации является основным компонентом школьного курса информатики. Однако многие методисты отмечают, что в настоящее время в практике преподавания именно алгоритмизации и программирования существует множество де-факто нерешенных проблем. В том числе исторически наблюдается сокращение фундаментальной подготовки в этой области в связи с увеличением технологической части курса. Сокращение предусмотрено и в нормативных документах, но на практике иногда наблюдается полное вытеснение алгоритмизации и программирования из базового курса информатики и ИКТ [1, 2].

В работе [3] разработаны подходы к обучению алгоритмизации и программированию в школе, в частности, отмечено, что наибольшее внимание следует уделить не только изучению алгоритмов с математической точки зрения, но и практическим аспектам их реализации на языке программирования [2]. В то же время важно и терминологическое разделение понятий «алгоритм» и «программа», а также развитие навыков перехода по необходимости перехода от одного языка программирования к другому [3], что особенно важно вследствие развития ряда сравнительно новых технологий программирования (специализированные скриптовые языки, компонентный подход и т. д.). Алгоритмическая подготовка ученика ведется в рамках базовой и профильной подготовки. Если в базовой части курса решается задача первоначального обучения основным алгоритмическим структурам, то в рамках профильных курсов целесообразно закреплять базовые алгоритмические навыки и способности к их применению при решении различных задач. В этой связи актуальным становится решение проблемы готовности учителей к ведению серьезных

профильных курсов в этом направлении, требующих высокой не только методической, но прежде всего и предметной квалификации.

В Томском государственном педагогическом университете проблема подготовки будущего учителя к такой деятельности решается путем внесения ряда специальных дисциплин в учебный план. Одна из таких дисциплин «Алгоритмы и структуры данных» [4] позволяет студентам получить практические навыки и теоретические знания в следующих областях: сортировка и поиск, методы разработки эффективных алгоритмов, кодирование информации и шифрование, алгоритмы на графах, а также способствует развитию навыков алгоритмического мышления и алгоритмической деятельности.

Целью данной дисциплины является обучение студентов основным алгоритмам в процессе практического решения задач на ЭВМ и развитие навыков эффективного программирования.

В работах [5, 6] показана важность развития алгоритмического мышления у студентов педагогических вузов как будущих учителей информатики. В работах [3, 7, 8] также представлена целесообразность введения в учебный план новых дисциплин, задачей которых является не только обучение конкретным технологиям, но и продолжение формирования алгоритмического мышления. К таким дисциплинам относится и дисциплина «Алгоритмы и структуры данных». В ее рамках изучаются так называемые базовые алгоритмы. Важность их изучения связана с тем, что, с одной стороны, эти алгоритмы активно используются при решении разнообразных задач на алгоритмизацию и программирование, с другой – на примере этих задач можно легко продемонстрировать базовые приемы разработки эффективных алгоритмов.

Целью данной работы является разработка методики обучения алгоритмам и структурам дан-

ных, направленной на развитие алгоритмического мышления.

На самом деле, разные исследователи по-разному определяют термин «алгоритмический стиль мышления». В данной работе будем опираться на те же понятия и определения, что и в работах [5, 6].

Поскольку алгоритмическая подготовка является наиважнейшей для будущего специалиста и без нее немислимо знанием предмета на приемлемом уровне, то предлагается направить на усиление этой подготовки резервы, предусмотренные дисциплинами, устанавливаемыми вузом. В ТГПУ в программу подготовки будущих учителей информатики введены такие дисциплины, как «Алгоритмы и структуры данных», «Вычислительная геометрия», «Объектно-ориентированное программирование», «Технология Java», «Web-программирование», «Трансляция с языков высокого уровня». Все эти дисциплины, помимо изучения соответствующих узких информационных технологий, направлены на формирование алгоритмических навыков и в этом плане дополняют базовые дисциплины «Программирование» и «Технологии программирования».

Содержание дисциплины определено рабочей программой [4]. В ее рамках последовательно изучаются следующие разделы – элементы теории эффективности, сортировка и поиск, элементы теории вычислимости, элементы теории формальных языков, NP-полные задачи и языки, элементы теории графов, рекурсивные алгоритмы, элементы теории информации.

Обучение предложенной дисциплины ведется по схеме, представленной на рисунке.



Методика обучения алгоритмам и структурам данных: БА – базовые алгоритмы; ТП – типовой пример; СР – самостоятельная работа (установочная часть)

Основная идея, реализуемая по ходу всей дисциплины, заключается в том, что в процессе изучения очередного алгоритма студент совершает ряд учебных действий.

В рамках первых тем алгоритм часто представляется студенту в виде, близком к итоговому тексту, реализующей его программы. От студента требуется совершить минимальный набор действий – оформить программу в соответствии с требуемой структурой, описать переменные и т. д. По-

степенно формальные описания алгоритмов становятся все менее детализированными, от студента требуется уже больше именно «программистских» навыков.

Этот подход согласуется с подходом поэтапного формирования алгоритмического мышления, рассмотренного в работах [5, 6].

Пример 1. Изучение алгоритма поиска в лабиринте (в процессе изучения особенностей рекурсивных алгоритмов).

Ниже представлен листинг рекурсивной процедуры.

```

1 procedure find(x,y: integer);
  {x и y – координаты ячейки, от которой осуществляется поиск}
2 begin
3   if (x=xk)and(y=yk) then begin
  {Проверяем, не является ли данная ячейка конечной.}
4     writeln;
5     writeln();
6     readkey
7     halt
  {Если проверка условия дала положительный результат, выдаем сообщение, что путь существует 'there is a way', и завершаем вызов процедуры. Если процедура завершится стандартным способом, значит, пути не существует, о чем и надо будет сообщить в основной программе.}
8   end;
9   lab[x,y]:=2;
  {Помечаем текущую ячейку как просмотренную – 2, что препятствует ее повторному просмотру во время поиска пути}
10  if lab[x+1,y]=1 then find(x+1,y)
  {Моделируем шаг вниз, если он возможен (ячейка свободна и не просмотрена) и вызываем процедуру поиска пути для данной ячейки. Аналогично на шагах 11, 12, 13, где моделируем шаги вверх, вправо и влево соответственно. Замечание: каждый вызов процедуры производится тогда и только тогда, когда предыдущие попытки вызова процедуры не были выполнены.}
11  if lab[x-1,y]=1 then find(x-1,y);
12  if lab[x,y-1]=1 then find(x,y-1);
13  if lab[x,y+1]=1 then find(x,y+1);
14  lab[x,y]:=1
  {Текущая ячейка помечается как свободная, что фактически моделирует шаг назад, для случая если мы зашли в «тупик» и дальнейшее движение не возможно.}
15 end.

```

Студенту уже дан алгоритм «поиска в лабиринте», и ему необходимо лишь задать исходные данные, описать переменные и вызвать процедуру поиска пути в лабиринте от начальной ячейки.

В следующем примере рассмотрим описание другой задачи, решаемой с помощью рекурсии.

Пример 2. Обобщенный алгоритм решения задачи о «стабильных браках».

Процедура Try (m) – алгоритм поиска супруги для мужчины m, поиск идет в порядке списка предпочтений мужчины m:

```
PROCEDURE search (m: man);
VAR r: rank;
BEGIN
FOR r := 1 TO n DO
выбор r-й претендентки для m;
IF допустимо THEN запись брака;
IF m – не последний THEN search (successor(m))
ELSE записать стабильное множество
END;
отменить брак
END
END
END Try.
```

Здесь задача перед студентом стоит более сложная, так как ему дается лишь обобщенный алгоритм и необходимо не только решить задачу о представлении переменных, но и дописать алгоритм, прежде чем приступить к выполнению программы.

В целом лабораторный практикум содержит следующие разделы:

– организация очереди и стека на массиве, организация очереди и стека с помощью динамического списка, организация односвязного динамического

списка, алгоритм Бауэра–Мура, метод Кнута–Морриса–Пратта, бинарные и сильноветвящиеся деревья;

– внутренняя и внешняя сортировки, простые алгоритмы сортировки массивов, улучшенные алгоритмы внутренней сортировки, алгоритмы внешней сортировки, алгоритмы поиска;

– методы симметричного и ассиметричного шифрования;

– рекурсивные алгоритмы: бэктрекинг и поиск с возвратом, поиск в лабиринте, задача о восьми ферзях, задача о стабильных браках;

– алгоритмы на графах: поиск остова графа, поиск транзитивного замыкания графа (алгоритм Уоршалла), поиск в ширину на графе, поиск в глубину на графе, поиск компонента связности графа, представление взвешенного графа в ЭВМ, поиск кратчайших путей в графе (алгоритм Дейкстры), поиск минимального остова (алгоритм Краскала).

Таким образом, в процессе работы решены следующие задачи:

1. Исследованы способы развития алгоритмического мышления в процессе обучения алгоритмам и структурам данных.

2. Разработано содержание дисциплины «Алгоритмы и структуры данных».

3. Разработан лабораторный практикум по дисциплине «Алгоритмы и структуры данных», направленный на развитие алгоритмического мышления.

Список литературы

1. Ильясов А. Н. К вопросу о роли и месте программирования в школьной информатике. URL: <http://www.rusedu.info/PrintArticle101.html> (дата обращения: 1.03.2015).
2. Лапчик М. П. и др. Методика преподавания информатики: учеб. пособие для студентов пед. вузов / под общ. ред. М. П. Лапчика. М., 2003, 624 с.
3. 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.
4. Рабочая программа дисциплины «Алгоритмы и структуры данных». URL: http://tspu.edu.ru/images2/fmf_news/UMKD/426/426_Б.3.В.06_Алгоритмы_и_структуры_данных.pdf (дата обращения: 1.03.2015).
5. 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.
6. 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.
7. 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.
8. 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.

Карташов Д. В., аспирант.

Томский государственный педагогический университет.

Ул. Киевская, 60, Томск, Россия, 634061.

E-mail: DeKar@tspu.edu.ru

Стась А. Н., зав. кафедрой, ведущий инженер-программист.

Томский государственный педагогический университет.

Ул. Киевская, 60, Томск, Россия, 634061.

E-mail: stasandr@tspu.edu.ru

Материал поступил в редакцию 05.06.2015.

D. V. Kartashov, A. N. Stas

METHODS OF TEACHING ALGORITHMS AND DATA STRUCTURES

The article is devoted to the methods of teaching students the basic often applied algorithms of solving computer problems. The main feature of this methodology is the consistent study of the basic algorithms (sorting and search, information theory and cryptography, recursive algorithms and graph algorithms) implying the use of any programming language and further implementation of the studied algorithms in practice. This course includes laboratory practical tasks. These tasks have the increasing difficulty. At the beginning of the course a simple task demands from the student filling of admissions. More difficult tasks assume independent development and debugging of computer programs.

Key words: *methods of teaching, algorithmic thinking, algorithms and data structures, sort and search, recursive algorithms and graph algorithms.*

References

1. Ilyasov A. N. *K voprosu o roli i meste programmirovaniya v shkol'noy informatike* [To the question on the role and place of programming in school computer science]. URL: <http://www.rusedu.info/PrintArticle101.html> (accessed 1.03.2015) (in Russian).
2. Lapchik M. P., Semakin I. G., Henner E. K. *Metodika prepodavaniya informatiki: ucheb. posobiye dlya studentov ped. vuzov* [The Technique of teaching of computer science: Study guide for the students of pedagogical higher schools]. Moscow, Akademiya Publ., 2003. 624 p. (in Russian).
3. Yakymenko 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).
4. *Rabochaya programma distsipliny "Algoritmy i struktury dannykh"* [The working program of the discipline "Algorithms and data structures"]. URL: http://tspu.edu.ru/images2/fmf_news/UMKD/426/426_Б.3.В.06_Алгоритмы_и_структуры_данных.pdf (accessed 1.03.2015) (in Russian).
5. Stas A. N., Dolganova N. F. *Razvitiye algoritmicheskogo myshleniya v protsesse obucheniya budushchikh uchiteley informatiky* [Algorithmic thinking development when training computer science teachers]. *Vestnik Tomskogo gosudarstvennogo pedagogicheskogo universiteta – TSPU Bulletin*, 2012, vol. 7 (122), pp. 241–244 (in Russian).
6. Stas A. N., Prusskikh 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).
7. 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).
8. 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).

Kartashov D. V.

Tomsk State Pedagogical University.

Ul. Kievskaya, 60, Tomsk, Russia, 634061.

E-mail: dekar@tspu.edu.ru

Stas A. N.

Tomsk State Pedagogical University.

Ul. Kievskaya, 60, Tomsk, Russia, 634061.

E-mail: stasandr@tspu.edu.ru