Информатика, компьютер, сложность вычислений

В. Н. Чубариков[1]

В работе обсуждаются актуальные проблемы информатики в целом и теории сложности вычислений в частности.

Ключевые слова: информатика, сложность вычислений, быстрое умножение

1. Программирование — основа информатики

Первый вопрос, который нам следует обсудить — "Что такое информатика?" Сказать, что "информатика" — это "компьютер-сайенс", что означает только отражение ее предмета, который является техническим устройством: вычислительной машиной, компьютером. Р. В. Хэмминг писал: "Мы называем наш предмет "информатикой", но мне кажется, что точнее было бы назвать его "компьютерной инженерией" (computer engineering), если бы не существовало вероятности неправильного толкования такого названия. Большей частью мы не подвергаем сомнению возможность существования **монитора, алгоритма, планировщика или компилятора**, скорее мы занимаемся поиском **практически работоспособного технического решения с разумными затратами времени и усилий**" ([1]).

Технический аспект здесь выступает на первое место в связи с тем, что большинство трудностей относится не к теоретическому обоснованию сделать что-то, а к практическому — каким образом это можно сделать проще и лучше.

Поэтому преподавание предмета "информатика" будет более эффективным, если в учебных планах дисциплина "программирование" будет предполагать в первую очередь практикум по программированию и не только, но и компьютерный практикум по разделам специализации, например, для учителей средней школы — компьютерный практикум по геометрии и по алгебре и началам анализа, и отводить для этого следует целый день занятий. Больше практических занятий! И в этом отличие в преподавании математики от преподавания информатики.

Чему учить? Первым выделим здесь языки и системы программирования, которые не являются прикладной составляющей обучения, и, сле-

[1] *Чубариков Владимир Николаевич* — заведующий кафедрой МКМА мех.-мат. фта МГУ, e-mail: chubarik2020@mail.ru.

Chubarikov Vladimir Nikolaevich — head of MCMA chair, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics

довательно, этому должен учить специалист по информатике. Заметим, что часто обучение ограничивается учебником грамматики и словарем (глоссарием) языка программирования. Но как полезно видеть и изучать, перенимать опыт подготовки добротных работающих программ. Следующий шаг — работа с базами данных требует уже участия специалистов, для которых интересна обработка этих данных. Здесь уже важна роль алгоритма и специальных законов конкретных наук, как социальных, гуманитарных, так и естественных. Заметим, что компьютер в настоящее время не обладает другими интеллектуальными возможностями кроме тех, которые присущи "цифровому вычислению". Тем не менее, в настоящий момент мы используем не все возможности компьютера как мощного инструмента управления и преобразования информации.

Наиболее содержательной частью информатики, являющейся предметом изучения и математики, являются численные методы. Поэтому этот материал в основном и представляется как теоретический, не подготовленный к практическому применению, в связи с недостаточной проработкой фундаментальных идей для решения задач. Зачастую, используя численные методы, компьютор позволяет нам разобрать достаточное множество частных примеров, чтобы выделить "модельные ситуации" того или иного явления, и даже если не удается сформулировать фундаментальных законов, но дает продвижение в познании явления. При этом нам приходится при составлении программ и планировании научной работы соизмерять соотношения между временем работы и памятью компьютора, между последовательными и параллельными вычислениями, между цифровыми и аналоговыми схемами и др.

Первоначальным понятием в информатике является понятие **алгоритма**. Оно определяется описательным образом словами разговорного языка. Алгоритм — точное предписание, которое задает вычислительный процесс (называемый алгоритмическим), начинающийся с некоторого исходного данного (совокупности возможных исходных данных) и направленный на получение результата, определяемого этим исходным данным. Под сложностью вычислений алгоритма понимают числовую функцию, оценивающую трудность применения алгоритма к исходным данным (время работы, число тактов работы при преобразовании исходных данных в заключительные и др.)

2. Модели вычислительных систем, компьютеры, языки программирования

Начнем изложение с определения машины Тьюринга (the Turing machine, 1936) — абстрактного вычислительного устройства, формально уточня-

ющего интуитивное понятие **алгоритма**. Она состоит из ленты, головки и управляющего устройства. Лента разделена на клетки и бесконечна влево и вправо. В каждой клетке ленты может быть записан только один символ из ленточного алфавита $A_0 = \{a_0, a_1, \ldots, a_k\}$, где a_0 — пустой символ. Головка машины может двигаться по ленте, перемещаясь из клетки в соседнюю клетку, читать символ, записанный в клетке, и записывать в обозреваемую клетку любой символ из A_0. Управляющее устройство перемещает головку по ленте и записывает символы в клетки. Оно может находиться в одном из состояний $q_0, q_1, \ldots, q_m$. Изменение положения головки и символов происходит по некоторой программе, состоящей из простейших команд.

Работа машины Тьюринга происходит в дискретном времени, начинается с исходных данных и завершается при достижении заключительного состояния (при некоторых исходных данных работа машины Тьюринга может и не заканчиваться). Кроме того, что машина Тьюринга дает точное определение вычислительного процесса, — алгоритма, она обладает наглядной реализацией алгоритмического процесса. Отметим, что машина Тьюринга является теоретической основой построения ЭВМ. Таким образом, любая машина Тьюринга с ленточным алфавитом производит алгоритмическое преобразование слов в этом алфавите.

Тезис Тьюринга. *Всякое реализуемое алгоритмическое преобразование можно выполнить подходящей машиной Тьюринга.*

3. Системы искусственного интеллекта

Специалист по информатике воспринимает свою главную функцию как обеспечение программ и ЭВМ для использования в старых и новых методиках обучения, но на нем лежит более сложная задача — выработка и распространение самого процесса обучения. Отправная точка зрения (М. Минский, С. Пайперт, 1969) этого мнения следующая.

1. Обучение языку программирования (хотя бы одному), работа со словарем этого языка.

2. Помочь людям строить в своем сознании различные виды вычислительных моделей.

3. Учитель должен иметь разумную модель того, что представляет собой сознание учащегося.

4. При отладке своих собственных моделей и процедур учащийся должен иметь модель того, что он делает и что он знает хорошие приемы отладки и простые, но решающие тестовые примеры.

5. Стремление учащегося при отладке программ узнать что-нибудь новое о вычислительных моделях и программировании в отличие от беспомощности представления о невозможности познать это.

Другими словами, учитель должен разрабатывать эффективные методы компьютерного моделирования процессов мышления, т.е. в определенном смысле работая с искусственным интеллектом.

Остановимся на универсальном решателе задач Ньюэлла и его коллег (General Problem Solver (GPS), 1957), исходной идеей которого являлось представление задач из некоторого класса как задач преобразования одного выражения в другое при помощи множества допустимых правил или, более общо, преобразованием одного состояния в другое. Добавим к этому использование общего механизма целенаправленного поиска для всех типов задач при изменении только конкретных знаний фактов и правил (базисные примеры логических формул).

Систематизацию и разработку решателя задач по элементарной алгебре и математическому анализу провел А. С. Подколзин ([2]). Он выделил три подхода компьютерного моделирования процессов решения задач.

Первый — древовидная классификация типов поддающихся алгоритмизации задач в соответствующей области и создание библиотеки процедур их решения (компьютерная алгебра, 1966).

Второй — основан на применении баз знаний, образованных аксиомами и теоремами некоторой предметной области (формальные языки, математическая логика, 1961).

Третий — использование базы алгоритмов локального планирования действий, накапливаемом при интерактивном обучении компьютерной системы, моделирующей процессы решения задач (решатель задач А. С. Подколзина, технология обучения, языки программирования, т.е. приемы решения задач).

4. Что такое TEX и LATEX?

Система компьютерной верстки, построенная на базе языка полиграфического оформления документов “TeX”, была создана Д.Кнутом (1979). Сила “TeX”а в упрощении работы пользователя и фактическом освобождении его необходимости программирования при верстке документов ([3]). Л.Лампорт (1984) представил систему “LaTex” ([4]). Существенным развитием ее стал “LaTeX2ε” (1994) с наборами пакетов расширений таких, как *beamer* — оформление презентаций, *AmS-Tex* — ввод математических формул, *XyMTeX* — ввод химических формул, *xypic* — построение диаграмм и т.п. ([5, 6]). Система “LaTeX2ε” нетребовательна к технике, не зависит ни от архитектуры компьютера, ни от установленной на нем операционной системы.

5. Сложность вычислений

Элементарной операцией назовем сумму или произведение двух цифр в двоичной системе счисления. Количество элементарных операций для сложения двух n-разрядных чисел есть $O(n)$, а для умножения в столбик — $O\left(n^2\right)$. А. Н. Колмогоров поставил задачу, что в этом смысле операция умножения сложнее сложения. Эта задача не решена до сих пор. Интуиция подсказывала А. Н. Колмогорову, что n^2 является оценкой снизу для количества элементарных операций. А. А. Карацуба опроверг это предположение.

Рассмотрим алгоритм А.А.Карацубы умножения многоразрядных (n-разрядных) чисел в двоичной системе счисления ([7]). Как известно обычный способ умножения чисел в столбик требует порядка n^2 элементарных "цифровых" операций. В алгоритме Карацубы достаточно использовать $n^{\log_2 3} \asymp n^{1.5}$ элементарных операций. Пусть перемножаются A и B — два $2n$-разрядных числа. Представим их в виде

$$A = 2^n A_1 + A_2, \quad B = 2^n B_1 + B_2,$$

где A_1, A_2, B_1, B_2 — n-разрядные числа. Имеем

$$AB = \left(2^{2n} - 2^n\right) A_1 B_1 + 2^n (A_1 + A_2)(B_1 + B_2) - (2^n - 1)\, A_2 B_2.$$

Следовательно, умножение $2n$-разрядных чисел сводится к умножению трех n-разрядных или $n+1$-разрядных чисел и нескольким операциям сложения и вычитания и сдвига чисел на не более $2n$ разрядов.

Если обозначить $M(n)$ количество элементарных операций для умножения двух n-разрядных чисел, то отсюда находим соотношение

$$M(2n) \leq 3M(n) + Cn,$$

где $C > 0$ — некоторая постоянная.

Следствием этого неравенства является оценка $M(n) \leq cn^{\log_2 3}$, где $c > 0$. В настоящее время Шенхаге и Штрассен построили алгоритм перемножения двух n-разрядных чисел с оценкой $M(n) \leq c_0 n \ln n \ln\ln n$, где $c_0 > c > 0$ — некоторые постоянные.

Алгоритм умножения двух квадратных матриц порядка n (умножение строки на столбец) требует примерно $n^2(2n-1)$ арифметических операций над элементами матриц. В. Штрассен (1970) предложил алгоритм умножения матриц за $O\left(n^{\log_2 7}\right)$, $\log_2 7 \asymp 2.807$. Пусть $AB = C$ — произведение двух матриц порядка $2k$. Тогда представим матрицы A, B, C в виде

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix},$$

где A_{ij}, B_{ij}, C_{ij}, $1 \le i, j \le 2$ — матрицы порядка k.

Имеем

$$\begin{aligned} C_{11} &= D_1 + D_4 - D_5 + D_7, \\ C_{12} &= D_3 + D_5, \\ C_{21} &= D_2 + D_4, \\ C_{22} &= D_1 + D_3 - D_2 + D_6, \end{aligned}$$

где

$$\begin{aligned} D_1 &= (A_{11} + A_{22})(B_{11} + B_{22}), \\ D_2 &= (A_{21} + A_{22})B_{11}, \\ D_3 &= A_{11}(B_{12} - B_{22}), \\ D_4 &= A_{22}(-B_{11} + B_{21}), \\ D_5 &= (A_{11} + A_{12})B_{22}, \\ D_6 &= (-A_{11} + A_{21})(B_{11} + B_{12}), \\ D_7 &= (A_{12} - A_{22})(B_{21} + B_{22}). \end{aligned}$$

Пусть $L(n)$ — число арифметических операций над элементами матриц в алгоритме Штрассена. Тогда из предыдущих соотношений находим

$$L(2n) \le 7L(n) + O\left(n^2\right).$$

Откуда следует, что $L(n) = O\left(n^{\log_2 7}\right)$, $\log_2 7 = 2.7807\ldots$.

Д.Копперсмит и С.Виноград (1990) уточнили этот до $O\left(n^{2.37}\right)$.

6. Поиск литературы по информатике

Приведем классификационную схему журнала ACM “Computing Reviews”.

C. Принципы построения компьютерных систем (архитектура процессоров, реализация компьютерных систем).

D. Программное обеспечение (методы программирования, разработка программного обеспечения, языки программирования, операционные системы).

F. Теория вычислений (вычисления посредством абстрактных устройств, анализ алгоритмов и сложность задач, логика и значение программ, математическая логика и формальные языки).

G. Математические вопросы теории вычислений (численный анализ, дискретная математика, теория вероятностей и математическая статистика).

H. Информационные системы (управление базами данных, хранение и поиск информации).

I. Методы вычислений (алгебраические манипуляции, искусственный интеллект).

J. Применения компьютеров (физические науки и инженерное дело).

К. Компьютеры и общество (история автоматизированных вычислений, компьютеры и образование, управление вычислительными и информационными системами, профессия программиста).

Список литературы

[1] Хэмминг Р. В., "Одна из точек зрения на информатику", *Лекции лауреатов премии Тьюринга: пер. англ.*, Мир, М., 1993, 240–254.

[2] Подколзин А. С., "О формировании приемов решения математических задач", *Интеллектуальные системы*, **3**:3–4 (1998), 51–74.

[3] Кнут Д. Е., *Всё про TeX*, РДТеХ, Протвино, 1993, 576 с.

[4] Львовский С. М., *Набор и вёрстка в системе LaTeX*, 3-е изд., испр. и доп., МЦНМО, М., 2003, 448 с.

[5] Есаян А. Р., Чубариков В. Н., Добровольский Н. М., Якушин А. В., *Подготовка документов в LaTeX2ε*, Уч.пос., Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, Тула, 2013, 390 с.

[6] Есаян А. Р., Чубариков В. Н., Добровольский Н. М., Якушин А. В., *Построение графиков средствами LaTeX-пакета pgfplots*, Уч.пос., Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, Тула, 2015, 372 с.

[7] Карацуба А. А., Офман Ю. П., "Умножение многозначных чисел на автоматах", *Доклады АН СССР*, **145**:2 (1962), 293–294.

Computer science, computer and computational complexity

Chubarikov V.N.

We discuss the problems of computer science and focus on computational complexity.

Keywords: computer science, computational complexity, fast multiplication

References

[1] Hamming R. W., "One Man's View of Computer Science", *Journal of the ACM*, **16**:1 (1968), 1–12.

[2] Podkolzin A. S., "Forming techniques for solving mathematical problems", *Intelligent Systems*, **3**:3–4 (1998), 51–74 (In Russian).

[3] Knuth D. E., *The TeXbook*, Addison-Wesley, 1986, 483 pp.

[4] Lvovskiy S. M., *Typesetting and layout in the LaTeX system*, 3rd eddition, MCCME, M., 2003 (In Russian), 448 pp.

[5] Esayan A. R., Chubarikov V. M., Dobrovolskiy N. M., Yakushin A. V., *Typesetting in LaTeX2ε*, Tula State Pedagogical University Publishers, Tula, 2013 (In Russian), 390 pp.

[6] Esayan A. R., Chubarikov V. M., Dobrovolskiy N. M., Yakushin A. V., *Plotting using LaTeX package pgfplots*, Tula State Pedagogical University Publishers, Tula, 2015 (In Russian), 372 pp.

[7] Karatsuba A. A., Ofman Yu. P., "Multiplication of many-digital numbers by automatic computers", *Doklady Akademii Nauk SSSR*, **145**:2 (1962), 293–294 (In Russian)