



В. Ф. Очков, А. В. Калинина,
 Национальный исследовательский университет «МЭИ», г. Москва

ПО ПОРЯДКУ СТАНОВИСЬ!

Аннотация

В статье рассмотрены вопросы последовательного и параллельного выполнения операторов при реализации алгоритма сортировки, а также создания программ сортировки вектора и анимации этой процедуры в среде Mathcad.

Ключевые слова: вектор, сортировка, анимация, Mathcad.

Контактная информация

Очков Валерий Федорович, доктор тех. наук, профессор Национального исследовательского университета «МЭИ», г. Москва; адрес: 111250, г. Москва, Красноказарменная ул., д. 14; телефон: (495) 362-71-71; e-mail: ochkov@twi.mpei.ac.ru

Калинина Алена Владимировна, студентка Национального исследовательского университета «МЭИ», г. Москва; адрес: 111250, г. Москва, Красноказарменная ул., д. 14; e-mail: kalinina.alena22@gmail.com

V. F. Ochkov, A. V. Kalinina,
 National Research University MPEI, Moscow,

THE ORDER GETS!

Abstract

The article discusses the issues of sequential and parallel execution of operators when implementing the sorting algorithm, as well as the creation of programs for sorting the vector and the animation of this procedure in Mathcad.

Keywords: vector, sorting, animation, Mathcad.

Как-то раз первый автор статьи разговорился со своей бывшей одноклассницей. Вспоминали хорошее и плохое школьных лет. Эта одноклассница призналась, что для нее одними из самых неприятных моментов в школе были... построения на уроках физкультуры. Девочки и мальчики строились в спортзале по росту и вставали друг против друга в две шеренги в стиле «белый верх — черный низ»: в белых маечках и черных трусиках. Эта одноклассница была самая высокая в классе и довольно крупная. В обычной жизни и в обычной одежде это было не так заметно, но в спортивном зале она оказывалась во главе шеренги полуголая (по стандартам тех времен) и как бы... у позорного столба. Мальчики ее разглядывали, а потом дразнили дылдой. Автор также вспомнил неприятный момент таких построений: у него был друг довольно низкого роста, который всегда оказывался в конце «мужской» шеренги, и на него из «женской половины» презрительно поглядывала девочка, которая ему нравилась*.

Да, нужно признать, что процедура построения по росту в школе или в армии несколько дискриминационная. В армии к тому же низкорослые солдаты оказываются в конце марширующей колонны и «глодают пыль». Можно предложить альтернативный порядок построения**, например, по возрасту, цвету глаз, волос или по какому-то другому атрибуту школьников или солдат, не имеющему признаков дискриминации. Но, так или иначе, тут нужно будет использовать процедуру *сортировки*, которая часто становится темой школьных уроков информатики. Операция вроде бы рутинная, разобранный «по косточкам», но давайте внесем в нее что-то новое.

Разработано множество алгоритмов сортировки. В «Википедии» можно найти такие их названия: сортировка пузырьком, сортировка перемешиванием, гномья сортировка, сортировка вставками, сортировка слиянием, сортировка с помощью двоичного дерева, сортировка Timsort, сортировка подсчетом, блочная сортировка, поразрядная сортировка, сортировка выбором, сортировка Шелла, сортировка расческой, пирамидальная сортировка, плавная сортировка, быстрая сортировка, интроспективная сортировка, терпеливая сорти-

* Говорят, что наш великий полководец Александр Суворов однажды выстроил своих крепостных по росту в две шеренги: неженатые пары с одной стороны, незамужние девицы — с другой. Так сложились пары, которым он повелел идти в церковь венчаться. Мнение новобрачных никого не интересовало. (https://lenta.ru/articles/2016/04/26/russian_slavery)

** Вообще, без построения по какому-либо признаку тут не обойтись: школьники или солдаты будут спорить и толкаться, стараться встать ближе к правому (приоритетному) краю шеренги или, наоборот, «затеряться в середине толпы». Можно предложить строить школьников в том порядке, в каком они по алфавиту записаны в классном журнале. Но тут также можно узреть элементы дискриминации: кого чаще вызывают к доске — Абрамова, Яковлева или, наоборот, Иванова-Петрова-Сидорова из середины списка. Кстати, к сортировке мы очень часто прибегаем, выстраивая файлы на экране компьютера по имени, типу, размеру или времени создания. Эти значения «в шеренге файлов» могут либо возрастать, либо убывать после повторного шелчка по имени атрибута.

ровка, рекурсивная сортировка, глупая сортировка, блинная сортировка, сортировка перестановкой*...

Эти алгоритмы реализованы на разных языках программирования. Попробуем и мы это сделать, но в среде пакета Mathcad [9, 11], который в настоящее время широко используется в школах и вузах**. Так, в списке использованных источников [1–3, 5, 8–12] приведены данные о статьях в журналах издательства «Образование и Информатика», где в названии присутствует слово Mathcad. Самих же статей, где речь идет об использовании этого пакета в учебном процессе, но где нет слова Mathcad в названии, гораздо больше.

Основа многих алгоритмов сортировки — это обмен численных значений у двух переменных. В языке программирования Basic, например, эту процедуру выполняет оператор swap(a, b). Но в математическом пакете Mathcad этого оператора нет, хотя его нетрудно реализовать другими средствами.

На рисунке 1 можно видеть два способа реализации оператора swap(a, b) в среде Mathcad: традиционный («треугольный» — с использованием вспомогательной переменной) и «экзотический», о котором многие пользователи Mathcad и не догадываются, — не через последовательное присваивание, а в некоем параллельном режиме, когда операторы присваивания записаны в матричном виде и выполняются независимо друг от друга. Эта технология обсуждалась на сайте пользователей Mathcad: <https://www.ptcusercommunity.com/message/448295>

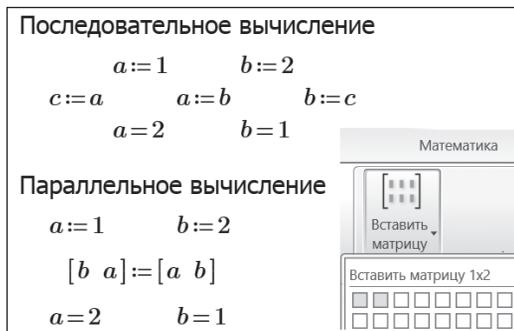


Рис. 1. Два способа перестановки значений переменных в среде Mathcad

* В студенческие годы первый автор подрабатывал на овощной базе. Самой непьющей и интеллигентной работой там считалась... сортировка лимонов. В те времена лимоны были единственными фруктами, какие продавались не по весу, а поштучно (к чаю) за 25, 30, 35, 40 и 45 копеек или что-то в этом роде: чем лимон крупнее, тем он, естественно, дороже. На овощных базах их сортировали так: просовывали в круглые дырки с разным диаметром на специальном сортировочном столе. Лимоны падали в отдельные ящики, которые потом развозили по разным магазинам, чтобы не было пересортицы. Автор же в те годы, глядя на эти ящики с отсортированными лимонами, вспоминал... вузовский курс математической статистики с его нормальным распределением: см. [6], а также рисунок 8, где можно увидеть нормальное распределение... студентов по росту. Анекдот тех времен: «Где подрабатываешь? — На овощной базе. — Устаешь? — Очень! — Ящики таскаешь? — Нет, лимоны сортирую — голова устаёт!»

** В пакете Mathcad есть, кстати, функция sort для этой работы, но мы пока забудем о ее существовании. Кроме того, встроенные в Mathcad функции csort и gsort позволяют сортировать матрицы по столбцам (c) или рядам (r), задавая ключ сортировки — номер столбца или строки матрицы. На сайте <https://www.ptcusercommunity.com/videos/6867> размещена анимация сортировки матрицы, когда значения ее элементов отображаются цветом.

Если под рукой есть оператор перестановки swap(a, b) в явном («треугольном») или в матричном виде, то несложно написать в среде Mathcad функцию-программу (рис. 2), реализующую, наверно, самый простой алгоритм сортировки, суть которого такова.

Школьники произвольно выстраиваются в ряд (у нас на рисунке 2 это делает функция runif, возвращающая вектор с десятью элементами и со случайными числами в интервал от 150 до 200), а учитель физкультуры (физрук) проходит вдоль шеренги слева направо (вложенный цикл for) и меняет местами пары школьников, если левый школьник (V_{i-1}) окажется выше правого (V_i). Эта процедура выполняется до тех пор, пока (while) очередной проход физрука вдоль шеренги не выявит беспорядка и переменная Flag (признак окончания сортировки) останется равной нулю.

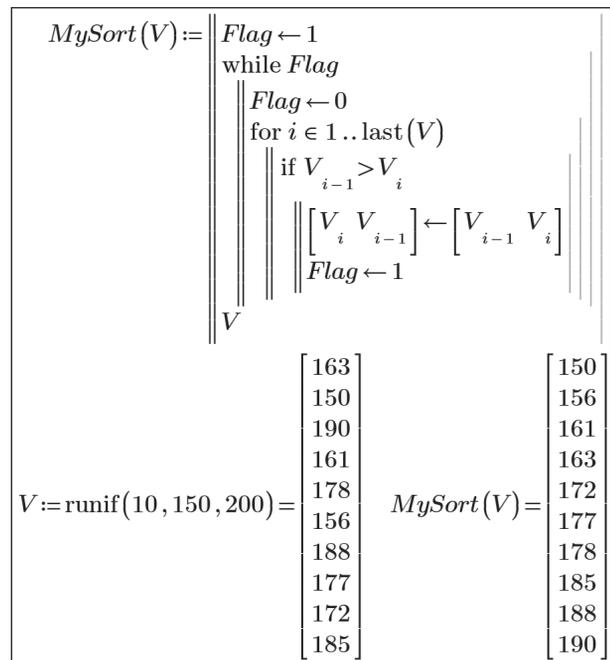


Рис. 2. Наипростейшая сортировка вектора (Mathcad Prime)

Пакет Mathcad 15 имеет удобные средства анимации [4, 7], которые можно применить и для визуализации сортировки векторов. На рисунке 3 показано, как в программу на рисунке 2 были вставлены три дополнительных оператора (они в рамках), позволившие показать сортировку вектора в анимации: столбики на диаграмме последовательно меняются местами, выстраиваясь в правильную шеренгу. При этом очередной «высокий столбик» перемещается вправо, как бы толкаясь и передавая эстафету столбику с более высоким ростом. Эта анимация выложена на сайте: <https://www.ptcusercommunity.com/videos/6795>

Управляет анимацией системная переменная FRAME, которая при создании анимации автоматически будет менять свое значение от 0 (рисует исходный вектор в виде столбиков) до 999 (отсортированный вектор, часть которого показана на рисунке 3). Перед созданием анимации можно вручную менять значение системной переменной FRAME и видеть изменения на графике — на столбчатой диаграмме. Так программа отлаживается перед анимацией. В программу на рисунке 3 вставлен счетчик перестановок соседних элементов векторов n . Если он становится равным номеру кадра анимации

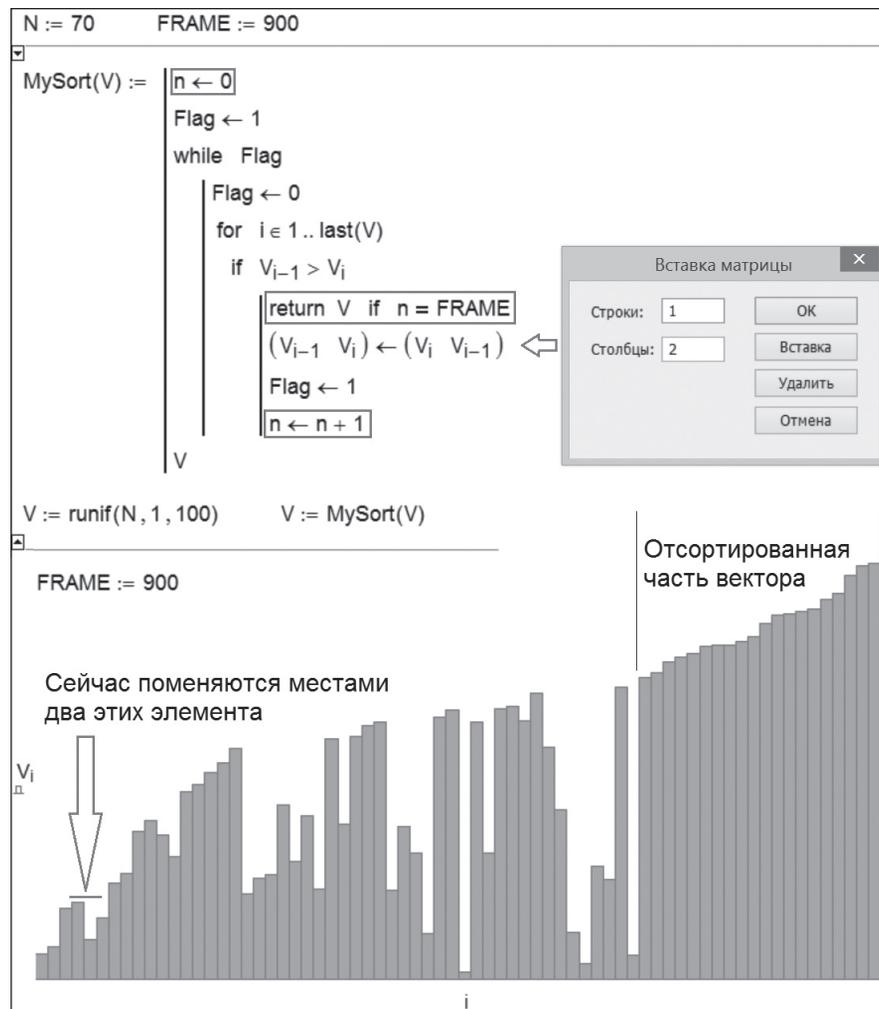


Рис. 3. Создание анимации сортировки по простейшему алгоритму (Mathcad 15)

FRAME, то программа оператором return возвращает частично отсортированный вектор, который рисуется на графике в виде столбчатой диаграммы.

Характер анимации, один кадр которой показан на рисунке 3, можно кардинально поменять, если в конце плеча оператора if (после оператора $n \leftarrow n + 1$) записать оператор break. Он прерывает выполнение цикла for и передает управление программой в его конец: физрук, переставив местами очередную пару школьников, не пойдет дальше с проверками вдоль шеренги, а пробежит в ее конец* и перейдет к циклу while. С оператором break отсортированные элементы вектора будут группироваться не на правом (рис. 3), а на левом фланге шеренги, но конечный результат будет тот же: слева окажутся самые низкие, а справа — самые высокие. Эта анимация размещена на сайте: <https://www.ptcusercommunity.com/videos/6824>

* В среде Mathcad, как и в языке Pascal, есть три «прерывающих» оператора: return, break и continue. Первый из них прерывает выполнение всей программы и возвращает из нее свой операнд (см. рис. 3). Второй оператор (он без операнда) прерывает выполнение цикла for или while и передает управление программой в конец цикла. Третий оператор тоже прерывает выполнение цикла, но передает управление не в его конец, в его начало. Кстати, мало кто даже из англичан знает, что означает break в слове завтрак — a breakfast. A fast — это по-английски «пост» (в смысле, не послание в Интернете, а ограничение в питании), а breakfast — это прерывание поста — разговение, которое имеет место на Пасху или в Рождество.

Программа сортировки на рисунке 3 имеет переменное число выполнения вложенных циклов while... for, и это число (оно хранится в переменной n) зависит от вида исходного вектора. Если этой программе «подсунуть» уже отсортированный вектор, то вложенный цикл while...for будет выполнен минимальное число раз, равное размеру вектора (у нас на рисунке 3 это 70 раз, как указано первым оператором). Если же этой программе дать уже отсортированный, но «перевернутый» вектор (этот «переворот» делает встроенная в Mathcad функция reverse), то вложенный цикл for будет «глупо»** выполняться максимальное число раз, переставляя пары элементов вектора. Эта анимация размещена здесь на сайте: <https://www.ptcusercommunity.com/thread/135862> как некий добавок к вопросу на форуме Mathcad по «перевертыванию» векторов.

Но в среде Mathcad несложно создать очень простую программу, сортирующую любой исходный вектор одним циклом. При этом число выполнения тела цикла будет равно числу элементов в исходном векторе (рис. 4).

Алгоритм в программе на рисунке 4 такой: физрук оценивает взглядом неотсортированную шеренгу школьников, находит самого высокого и отправляет его

** Если «глупой» обезьяне дать очищенный банан, то она попытается его еще раз очистить. Так и наша программа на рисунке 3 будет еще раз пытаться отсортировать уже отсортированный вектор.

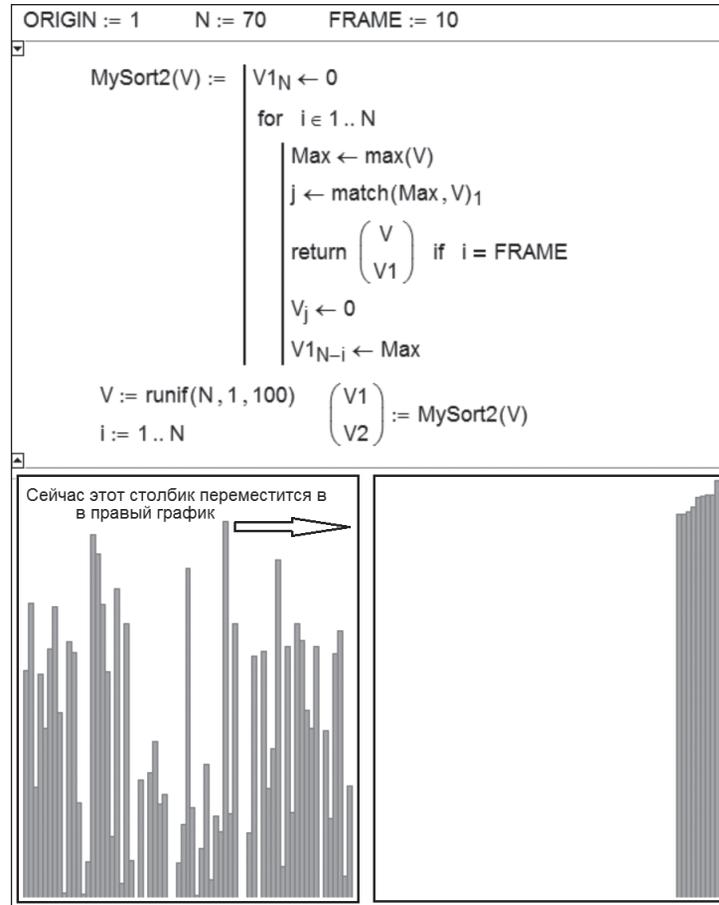


Рис. 4. Создание анимации сортировки по второму алгоритму (Mathcad 15)

на левый фланг отдельной формируемой шеренги. Эта операция повторяется N раз, где N — число элементов в исходном векторе. На столбчатых диаграммах рисунка 4 видно, что из исходного вектора в формируемый вектор перенесено девять «высокорослых» элементов (см. пустые места в левой диаграмме) и очередной переноса ожидает очередной «высокорослый» элемент. Программа на рисунке 4 исполняет этот простой алгоритм, работая с двумя специфическими функциями пакета Mathcad:

- `max` — определение максимального элемента массива (в нашем случае вектора);
- `match` — определение номера или номеров элементов массива с заданным значением (эта функция выдает вектор с одним или более элементами, поэтому-то в программе на рисунке 4 эта функция записана с единичным индексом).

Примечание. Самый «левый» элемент вектора в среде Mathcad по умолчанию имеет нулевой номер. Это умолчание присутствовало в программах на рисунках 2 и 3. Но в программе на рисунке 4 это умолчание нарушено: второй важной системной переменной Mathcad — переменной ORIGIN — присваивается единица. После этого нумерация векторов будет начинаться с единицы*. Но в программировании тут более привычен нуль. И еще. Алгоритм, реализованный в программе на рисунке 4, применим к исходному вектору с положительными

* Программиста призвали в армию, поставили в начале шеренги и отдали команду: «По порядку рассчитайсь!» Программист сначала уточнил, по какой системе (двоичная, восьмеричная, десятичная, шестнадцатеричная или какая-то другая особая) нужно вести счет, а потом выкрикнул: «Нулевой!»

элементами. Но его несложно доработать и для вектора с любыми числовыми элементами.

Можно развить алгоритм, реализованный в программе на рисунке 4, — выбирать из неотсортированной части вектора не только максимальный, но и минимальный элемент и отправлять его на левый фланг формируемого нового вектора.

Анимация, создание которой отображено на рисунке 4, расположена по адресу: <https://www.ptcusercommunity.com/videos/6837>.

Школьники могут попробовать реализовать в среде Mathcad и, главное, анимировать алгоритмы сортировки, описанные в «Википедии» (см. выше) или свои собственные. При этом можно использовать встроенную в Mathcad функцию `time` для оценки скорости работы тех или иных пользовательских функций сортировки по сравнению со встроенной функцией `sort`. При этом нужно иметь в виду, что скоростные изошренные алгоритмы сортировки создавались во времена тихоходных вычислительных машин, когда время счета было одним из главных лимитирующих факторов. Теперь же право на жизнь имеют и медленные алгоритмы, у которых их суть лежит на поверхности — как в наших программах.

Есть такая наука бионика. Она позволяет решать технические проблемы, наблюдая за поведением живых организмов. Например, один из самых оптимальных алгоритмов решения задачи коммивояжера [5] подсказали... муравьи, притаскивающие грузы в муравейник наикратчайшим маршрутом. Мы, создавая алгоритм сортировки, тоже можем обратиться к «живым организмам» — построить в спортзале шеренгу школьников,

например, по алфавиту, а затем приказать им перестроиться по росту. Все это можно записать на видеокамеру, а затем с помощью замедленного просмотра на мониторе компьютера проанализировать и попытаться воссоздать данный «живой» алгоритм самосортировки. Если сортировку, проводимую физруком (см. выше), можно реализовать на однопроцессорной машине, то самосортировка школьников потребует уже многопроцессорной системы. При этом физрук может вмешиваться в процесс такой сортировки, контролировать его. Тут на память приходят реальные технические объекты или объекты живой природы. В современном самолете есть и центральный компьютер, и локальные вычислительные устройства (контроллеры), один из которых, например, регулирует подачу топлива в двигатель, прекращает подачу топлива при нештатной ситуации. У человека есть «центральный» мозг, но человек отдергивает руку от горячего предмета рефлекторно, без задействования «умственных способностей». Все это повышает надежность работы сложных технических и живых систем, повышает их управляемость.

И вторая, более простая тема для исследования. Как изменится характер сортировки, если в конце плеча альтернативы программы на рисунке 2 поставить не оператор break, а оператор continue? И вообще, стоит ли вставлять их в программу? Как это будет влиять на скорость сортировки исходных векторов с разным распределением векторов «по росту»?

Статистическая добавка к статье.

На рисунке 2 показано, как встроенная в Mathcad функция `runif` генерирует вектор, содержащий случай-

ные числа в интервале от 150 до 200. Числа 150 и 200 — это условно минимальный и максимальный рост человека в сантиметрах. При этом было сделано допущение, что процент людей с определенным диапазоном роста одинаков. На рисунке 5 показана гистограмма (частота встречи) чисел, генерируемых функцией `runif`. При этом мы отказались от сантиметров и перешли к старым русским аршинам и вершкам и допустили, что рост человека может меняться от двух аршин до сажени (три аршина). Почему мы отказались от сантиметров?

Когда-то перед лекцией по регрессионному анализу в курсе «Информационные технологии» первый автор статьи подбирал пример статистической выборки для такого исследования. Но когда он «взошел на кафедру» и взглянул на аудиторию, то понял, что эта выборка находится прямо перед его глазами. Была проведена переключка студентов. Юноши вставали и сообщали свой вес и рост. Студентки, одной из которых была второй автор статьи, по понятным причинам, были освобождены от этой процедуры. Числа заносились в два вектора с именами «Вес» и «Рост» в среде математической программы Mathcad. В векторах получилось по 50 элементов. Эти массивы чисел послужили хорошей затравкой для лекции и основой для построения графика на рисунке 6 и вывода соответствующей регрессионной линейной зависимости веса, вернее, массы и роста юношей.

Захотелось также построить гистограмму роста студентов. Но один сантиметр в качестве элементарного диапазона роста тут не подошел: никакой статистической зависимости из такой гистограммы получить не удалось. Не подошли тут и английские футы с дюй-

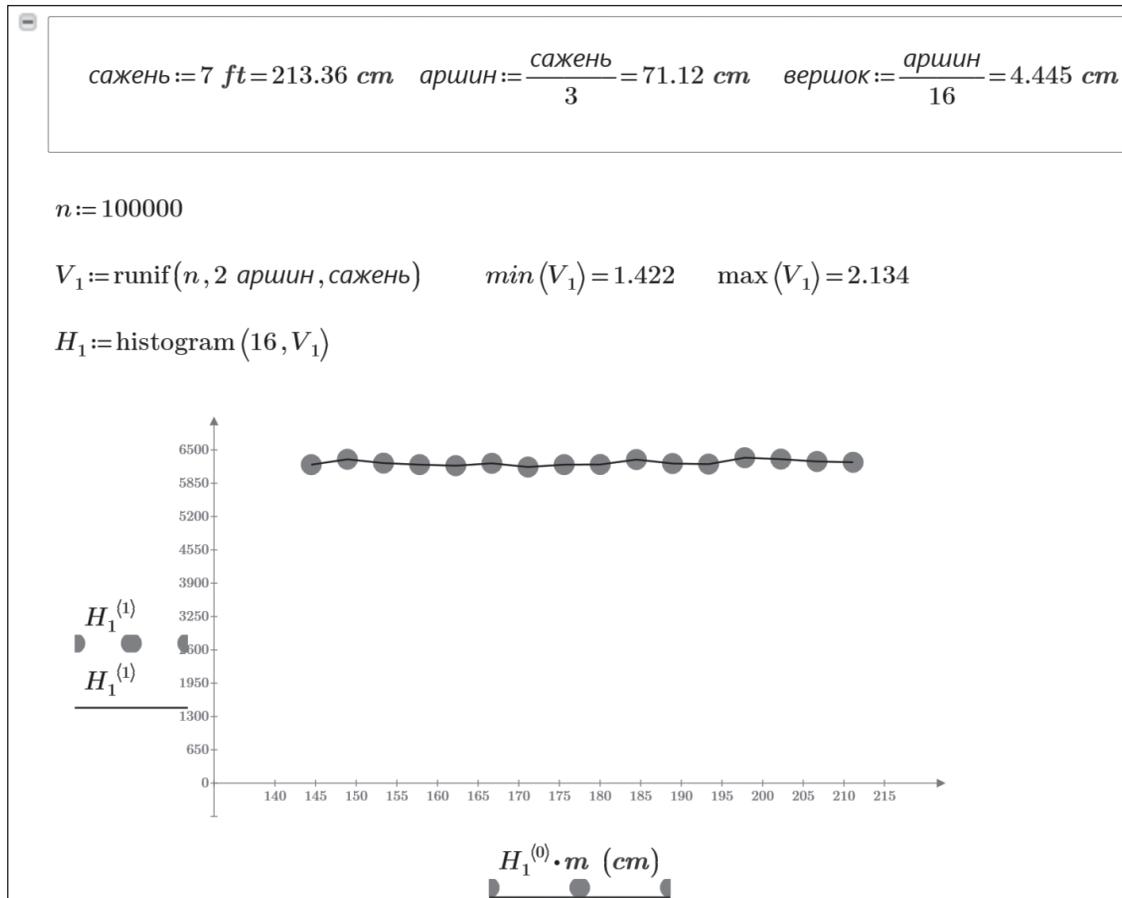


Рис. 5. Гистограмма равномерного распределения

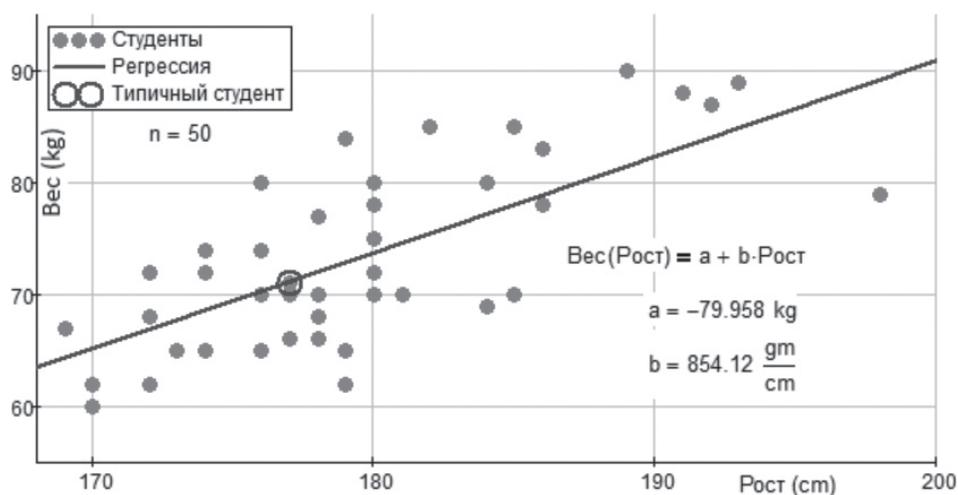


Рис. 6. Статистическая обработка параметров студентов

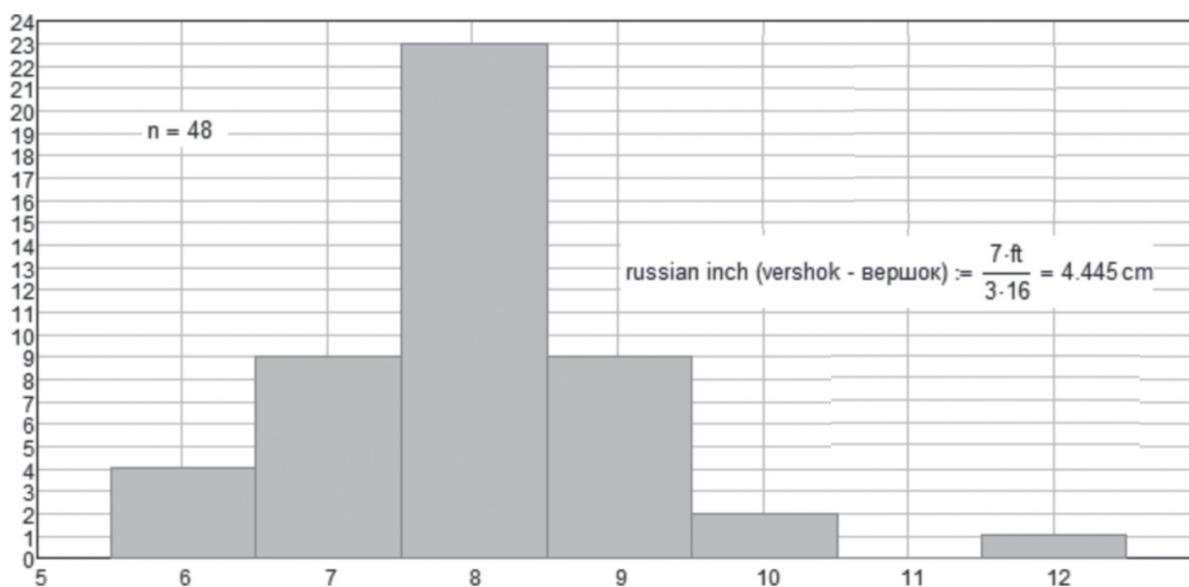


Рис. 7. Гистограмма роста студентов

мами*. Но старые добрые русские вершки оказались что надо!

На «вершковой» гистограмме (рис. 7) четко просматривается некий «статистический колокол» — кривая нормального распределения. Молодые люди по росту разбиваются на пять условных групп:

- низкого роста (четыре человека с шестивершковым ростом);
- скорее низкого, чем среднего роста (девять человек — семь вершков);
- среднего роста (23 человека — восемь вершков);
- скорее высокого, чем среднего роста (девять человек — девять вершков);
- высокого роста (два человека, 10 вершков).

Один студент выпал из нашей «статистики» и оказался ростом с тургеневского Герасима из повести «Муму»

* В одном из романов Агаты Кристи можно прочитать, что рост Эркюля Пуаро равнялся пяти футам и пяти дюймам. Такой же рост имел предок Форсайтов, описанный в знаменитой «Саге» Джона Голсуорси. Знание старинных и иностранных единиц измерения помогает лучше чувствовать «дух» классической литературы.

в 12 вершков**: исключение только подтверждает статистическое правило.

В Интернете можно найти не очень качественную, но очень интересную фотографию «живой» гистограммы роста людей (рис. 8): «девочки в белом, мальчики в черном».

Группа игроков выстроилась на поле по росту, но не в одну шеренгу, как на уроках физкультуры (см. выше), а в колонну с разным числом людей в каждом ряду. Каждый ряд соответствовал определенному росту: пять футов (152,4 см), пять футов и один дюйм (154,94 см) и так далее до 6 футов и 5 дюймов (195,58 см). В среде Mathcad есть функция `pnorm`, позволяющая генерировать случайные числа с нормальным (рис. 9), а не с равномерным (см. рис. 5) распределением. Подбирая значения первого (математическое ожидание) и второго (среднеквадратическое отклонение) аргументов этой функции, можно

** Читаем у Тургенева: «Из числа всей ее челяди самым замечательным лицом был дворник Герасим, мужчина двенадцати вершков роста, сложенный богатырем и глухонемой от рождения». Рост Герасима был равен двум аршинам и 12 вершкам, но аршины не указывали по умолчанию.

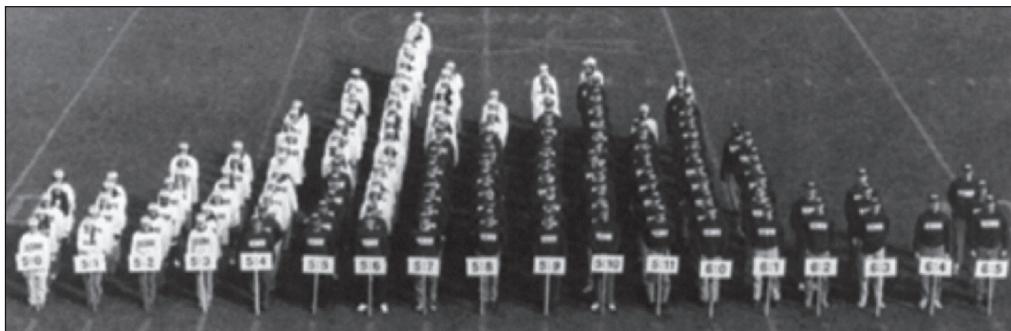


Рис. 8. «Живая» гистограмма роста человека

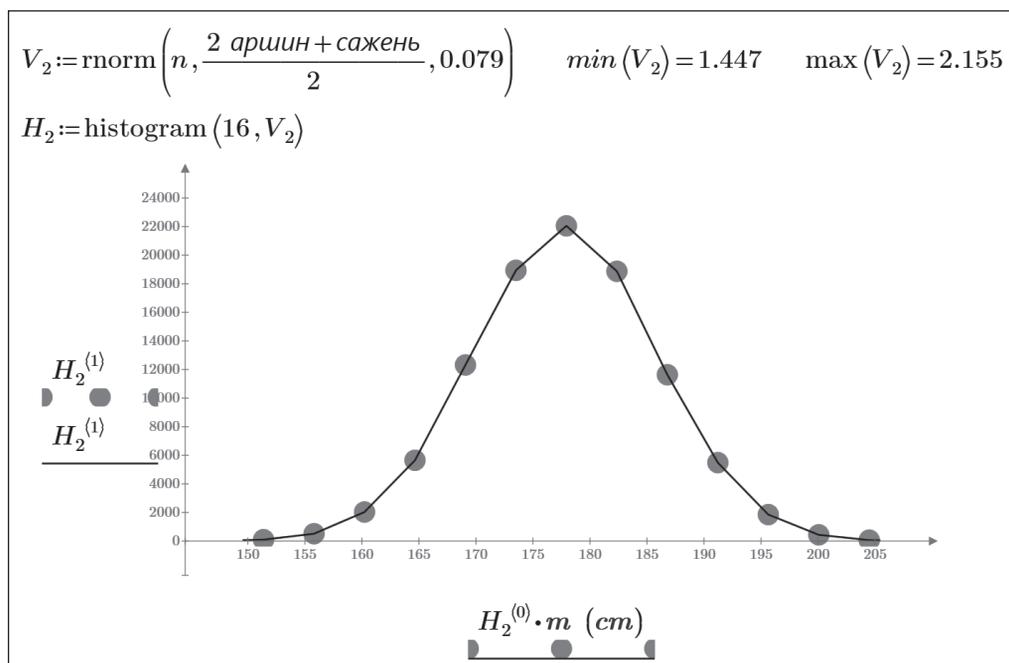


Рис. 9. Генерация вектора с нормальным распределением в среде Mathcad

добиться требуемого распределения студентов по росту. Но на сортировку характер распределения школьников, студентов, солдат по росту не влияет.

Список использованных источников

1. Андрафанова Н. В., Попова Г. И. Использование возможностей пакета Mathcad для генерации и проверки индивидуальных заданий // Информатика и образование. 2014. № 10.
2. Бгатова О. В. Применение Mathcad при обучении математике в колледже // Информатика и образование. 2007. № 11.
3. Очков В. Ф. Mathcad и криптография // Информатика в школе. 2013. № 10. <http://tw.t.mpei.ac.ru/ochkov/Mathcad-15/MATHCAD-CRYPTOGRAPHY.pdf>
4. Очков В. Ф. Живые кинематические схемы в Mathcad // Открытое образование. 2013. № 3. <http://tw.t.mpei.ac.ru/ochkov/Mathcad-15/kinematic.html>
5. Очков В. Ф. Пойти туда, зная куда // Информатика в школе. 2014. № 10. <http://tw.t.mpei.ac.ru/ochkov/Comm-i-maket.pdf>
6. Очков В. Ф., Богомолова Е. П. Интерполяция, экстраполяция, аппроксимация, или Ложь, наглая ложь и стати-

стика // Cloud of Science. 2015. Т. 2. № 1. <http://tw.t.mpei.ac.ru/ochkov/stat.html>

7. Очков В. Ф., Богомолова Е. П., Иванов Д. А. Физико-математические этюды с Mathcad и Интернет: учебное пособие. СПб.: Лань, 2016. <https://www.ptcusercommunity.com/groups/etudes>

8. Очков В. Ф., Чудов В. Л., Соколов А. В. Использование форума РТС Community/Mathcad на школьных занятиях по информатике // Информатика в школе. 2015. № 10. <http://tw.t.mpei.ac.ru/ochkov/RTC-Community-Lyc.pdf>

9. Суханов М. Б. Применение пакета Mathcad в обучении программированию на языках высокого уровня // Информатика и образование. 2011. № 10.

10. Черняк А. А., Черняк Ж. А., Якимович А. А. Mathcad и Excel для школьников: решение уравнений и неравенств // Информатика и образование. 2009. № 3–7.

11. Шамсутдинова Т. М. Програмируем в системе Mathcad // Информатика и образование. 2006. № 5.

12. Шушкевич С. В. Обучение построению графических объектов в Mathcad // Информатика и образование. 2009. № 5, 6.