



И. А. Тюшнякова,

Таганрогский институт имени А. П. Чехова (филиал)

Ростовского государственного экономического университета (РИНХ)

ИЗУЧЕНИЕ СОРТИРОВКИ ПОДСЧЕТОМ И СЛИЯНИЕМ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

Аннотация

Статья посвящена вопросам изучения сортировки в школьном курсе информатики. Изложены модификации методов сортировки подсчетом и слиянием, характеризующиеся визуально понятным и наглядным представлением алгоритмов. Описана возможность познакомить учащихся с основами параллельного программирования при изучении данных методов сортировки.

Ключевые слова: сортировка подсчетом, сортировка слиянием, алгоритм, программирование.

DOI: 10.32517/2221-1993-2018-17-7-32-35

Контактная информация

Тюшнякова Ирина Анатольевна, канд. тех. наук, доцент, доцент кафедры информатики Таганрогского института имени А. П. Чехова (филиала) Ростовского государственного экономического университета (РИНХ); адрес: 347936, Ростовская область, г. Таганрог, ул. Инициативная, д. 48; телефон: (8634) 60-18-07; e-mail: Solovyova_Irina@mail.ru

I. A. Tyushnyakova,
Taganrog Institute named after A. P. Chekhov
(branch) of Rostov State University of Economics

STUDYING THE COUNTING SORT AND MERGE SORT IN THE SCHOOL COURSE OF INFORMATICS

Abstract

The article is devoted to the study of sorts in the school course of informatics. Modifications of the sorting and merging methods are described, characterized by a visually understandable and visual representation of the algorithms. An opportunity to acquaint students with the basics of parallel programming in the study of sorting data is described.

Keywords: counting sort, merge sort, algorithm, programming.

Построение алгоритмов, изучение языков программирования являются неотъемлемыми частями освоения курса информатики. В процессе программирования у учащихся развивается логическое и алгоритмическое мышление, вырабатываются творческие способности. Авторские программы по информатике и ИКТ для среднего общего образования различных авторов предполагают изучение алгоритмов сортировки в рамках раздела «Алгоритмы и элементы программирования».

Сортировка относится к одной из базовых процедур нечисловой обработки данных и заключается в упорядочении элементов в списках [1, 2]. Алгоритмы сортировки имеют достаточно широкий спектр применения и используются в разнообразных задачах, которые связаны с системами автоматизированного управления (САУ), информационно-поисковыми системами и т. д. Сортировка может применяться для эффективной обработки в больших наборах данных (например, для поиска); для представления массивов данных в удобной для анализа форме; с целью группировки элементов по определенному признаку и т. п. В работах [3, 4] нами было рассмотрено использование алгоритмов сортировки для решения задач вычислительной математики и распознавания изображений, в частности, подробно (с листингами и результатами численных экспериментов) были описаны авторские схемы применения сортировок для нахождения экстремумов, корней и полюсов функций с разнообразными приложениями.

В школьном курсе информатики учащихся знакомят, как правило, с наиболее простыми (в плане программной реализации) и известными алгоритмами сортировки: методом выбора, методом обмена («пузырька»), быстрой сортировкой.

В качестве одного из интересных алгоритмов можно отметить один из частных случаев класса сортировки *m*-путевым слиянием — сортировку подсчетом.

Суть **сортировки подсчетом** состоит в определении номеров мест элементов массива, расстановка по которым даст отсортированный массив. Метод основан на том, что *j*-ключ упорядоченной последовательности больше (*j* – 1)-ключа массива. Сортировка подсчетом представляется с помощью **матрицы сравнений**. Модификация метода, характеризующаяся визуально понятным, наглядным представлением алгоритма, разработана доктором технических наук, профессором Я. Е. Роммом [5].

Рассмотрим массив неупорядоченных чисел $A = (1, -5, 7, -1, 4, -5)$, состоящий из $N = 6$ элементов, расположим его в первой строке и в первом столбце матрицы сравнений (рис. 1).

Матрица сравнений заполняется знаком разности элементов, расположенных горизонтально в строке и вертикально в столбце. С целью определения места элемента в отсортированном массиве следует просуммировать число нулей и плюсов в каждом столбце матрицы сравнений над диагональю (вместе с диагональным элементом) и сложить получившееся

	1	-5	7	-1	4	-5
1	0	-	+	-	+	-
-5	+	0	+	+	+	0
7	-	-	0	-	-	-
-1	+	-	+	0	+	-
4	-	-	+	-	0	-
-5	+	0	+	+	+	0
	↓	↓	↓	↓	↓	↓
	4	1	6	3	5	2

Рис. 1. Матрица сравнений для массива $A = (1, -5, 7, -1, 4, -5)$

число с количеством только плюсов ниже диагонали. Полученные значения запоминаются в счетчик k , значение k получается адресом вставки $A1[k] = A[j]$, при этом можно запомнить и номер элемента в исходном массиве (входной индекс вставленного элемента) $e[k] = j$. На выходе сортировки $A1[k]$ и $e[k]$ имеют равный номер k .

На рисунке 1 подсчитанные номера элементов в отсортированном массиве представлены ниже таблицы цифрами рядом со стрелками. Структурная схема алгоритма представлена на рисунке 2.

Программа, реализующая сортировку подсчетом на языке Pascal, представлена на рисунке 3.

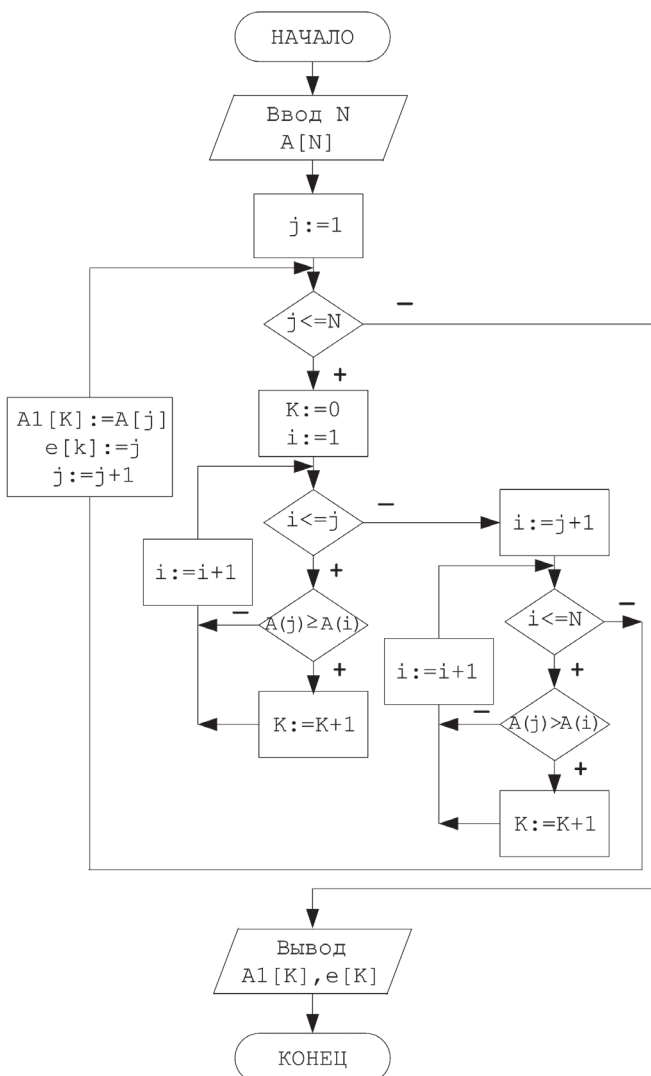


Рис. 2. Структурная схема алгоритма сортировки подсчетом

```

program Sort_podchet;
const N=6;
A: array[1..6] of integer=(1, -5, 7, -1, 4, -5);
var i,j,k:integer;
A1,e: array[1..N] of integer;
begin
for j:=1 to N do
begin
k:=0;
for i:=1 to j do if A[j]>=A[i] then k:=k+1;
for i:=j+1 to N do if A[j]>A[i] then k:=k+1;
A1[k]:=A[j]; e[k]:= j;
end;
writeln('Отсортированный массив:');
for i:=1 to N do
Write(A1[i]:5);
writeln;
writeln('Номера элементов в исходном массиве:');
for i:=1 to N do
Write(e[i]:5);
end.
  
```

Рис. 3. Реализация сортировки подсчетом в среде PascalABC.NET

Данная сортировка допускает эффективное распараллеливание с единичной оценкой временной сложности. Эта оценка получается из следующих соображений. На основании взаимной независимости элементов матрицы сравнений можно установить знак всех ячеек матрицы за время одного сравнения. В силу известной диагонали (с нулевыми элементами) и симметричности матрицы сравнений для этого потребуется количество процессоров $\frac{N^2 - N}{2} \leq \frac{N^2}{2}$; время выполнения шага

$O(1)$, т. е. $T\left(\frac{N^2}{2}\right) = O(1)$, где N — число сортируемых элементов.

Помимо сортировки подсчетом интерес может представлять **сортировка слиянием**. Она основана на адресном слиянии двух ранжированных массивов по матрице сравнений.

В качестве примера на рисунке 4 приведена матрица сравнений для двух сливаемых массивов $a = (2, 3, 13, 14)$ и $b = (3, 4, 4, 5)$. Массив a располагается в матрице сравнений горизонтально сверху, массив b — вертикально

	$-\infty$	2	3	13	14	∞
$-\infty$	0	+	+	+	+	+
3	-	-	0	+	+	+
4	-	-	-	+	+	+
4	-	-	-	+	+	+
5	-	-	-	+	+	+
∞	-	-	-	-	-	0

Рис. 4. Матрица сравнений для слияния двух массивов с указанием направления обхода

слева, к матрице сравнений добавляются окаймляющие строки и столбцы. Ячейки матрицы сравнений заполняются аналогично вышеописанной сортировке подсчетом.

Слияние двух массивов a и b организуется путем поэлементного обхода слева направо и сверху вниз по неотрицательной части матрицы сравнений. Направление обхода указано стрелками на рисунке 4. Выходной массив c сформируется в соответствии со следующими соотношениями:

$$c_{i+j} = \begin{cases} b_i, & \text{если } \alpha_{ij} = -1, \alpha_{i(j+1)} \geq 0, \\ a_j, & \text{если } \alpha_{ij} \geq 0, \alpha_{(i+1)j} = -1, \end{cases} \quad \alpha_{ij} = \begin{cases} 1, & a_j > b_i, \\ 0, & a_j = b_i, \\ -1, & a_j < b_i. \end{cases}$$

Здесь:

i и j — произвольно взятые индексы при ограничениях $0 \leq i \leq n+1, 0 \leq j \leq m+1$;

n и m — количество элементов в ранжированных массивах b и a ;

α_{ij} — элемент i -й строки и j -го столбца матрицы сравнений.

Программная реализация слияния двух упорядоченных массивов на языке Pascal представлена на рисунке 6.

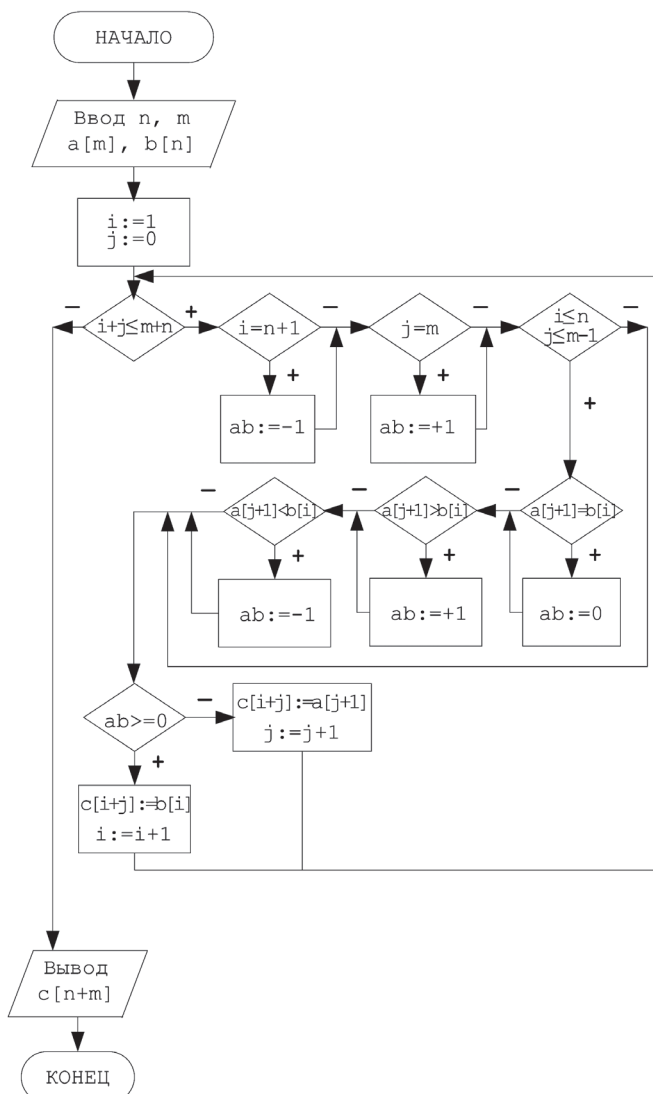


Рис. 5. Структурная схема слияния двух упорядоченных массивов

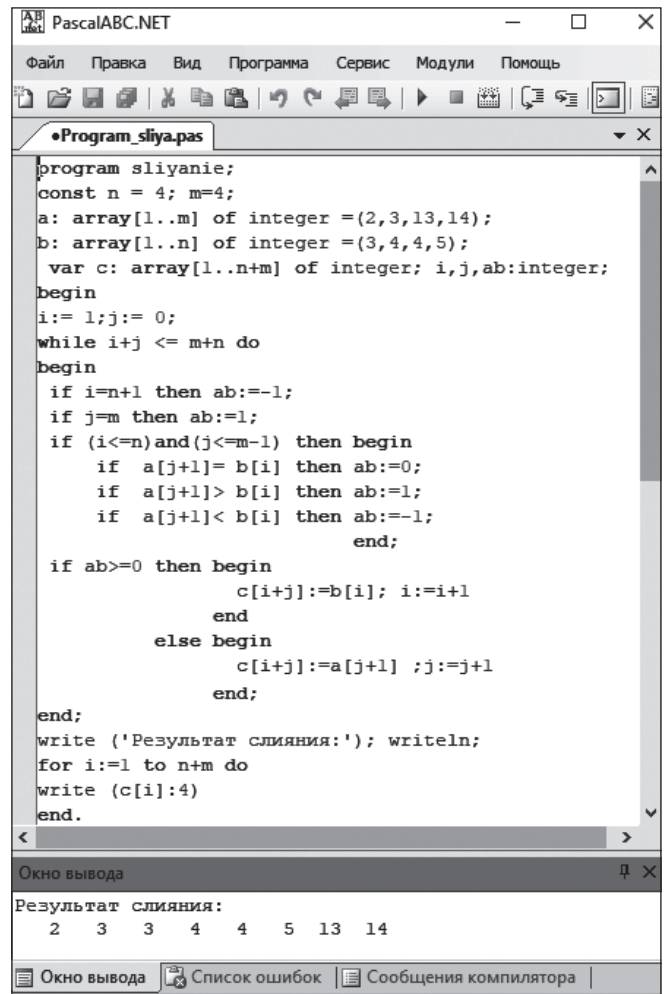


Рис. 6. Реализация слияния двух массивов в среде PascalABC.NET

Сортировка слиянием представляется в виде рекурсивного алгоритма, выполняющего свою работу, двигаясь вверх по рекурсии. При этом исходный неупорядоченный массив необходимо рассматривать как множество, состоящее из N подмножеств, каждое из которых состоит из одного элемента. Слияния начинаются с подмножеств, состоящих из одного элемента. Парно сливаются смежные подмножества, образуя подмножества из двух элементов, и так далее до тех пор, пока не образуется одно отсортированное множество. Количество подмножеств на каждом этапе слияния уменьшается вдвое, а их размеры удваиваются. Кратко охарактеризовать сортировку слиянием можно следующим образом: первоначально производится слияние каждой пары элементов с запоминанием обратных адресов, затем слияние упорядоченных пар, затем упорядоченных четверок и т. д. Полный листинг программы сортировки слиянием представлен в [3].

Время выполнения максимально *параллельно*-го варианта сортировки слиянием оценивается как $T\left(\frac{N^2}{4}\right) = O(\log_2 N)$, где $\frac{N^2}{4}$ — количество процессоров. *Последовательный* вариант сортировки слиянием выполняется за время $O(\log_2 N)$ [4]. Первая оценка получается из следующих соображений. Одно слияние на i -м шаге сортировки получается за время одного сравнения τ в силу взаимной независимости элементов матрицы сравнений. Такое слияние требует $(2^{i-1})^2$ про-

цессоров. Число взаимно независимых слияний на i -м шаге равно $\frac{N}{2^i}$, откуда число процессоров на шаге — $\frac{N}{2^i} \cdot (2^{i-1})^2 \leq \frac{N^2}{4}$; время выполнения шага $O(1)$. С учетом числа шагов $\log_2 N$ получается представленная оценка временной сложности параллельной сортировки слиянием.

Алгоритмы сортировки подсчетом и слиянием отличаются от известных аналогов параллелизмом, устойчивостью (сохранением порядка равных элементов), а также прямой и обратной адресностью.

Помимо визуально интересного представления алгоритмов сортировки подсчетом и слиянием их изучение в рамках уроков информатики либо факультативных или элективных курсов дает возможность познакомить учащихся с основами параллельного программирования. Представленные оценки временной сложности параллельного выполнения сортировки подсчетом и слиянием показывают возможность эффективного распараллеливания сортировки с помощью компьютер-

ных MIMD-систем*, когда мы имеем дело с несколькими процессорами, каждый из которых способен выполнять свою инструкцию, при этом существует несколько потоков данных и каждый процессор может работать со своим набором.

Список использованных источников

1. Вирт Н. Алгоритмы и структуры данных. М.: ДМК-Пресс, 2016.
2. Кнут Д. Искусство программирования. В 3 т. Т. 3. Сортировка и поиск (второе издание). М.: Вильямс, 2017.
3. Ромм Я. Е., Тюшнякова И. А. Применение сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений: учебное пособие. Таганрог: Изд-во Таганрог. гос. пед. ин-та, 2009.
4. Тюшнякова И. А. Разработка и исследование схем применения сортировки для поиска нулей и особенностей функций с приложением к идентификации плоских изображений: дис. ... канд. тех. наук. Таганрог, 2006.
5. Romm Ya. E. Sorting-based calculation of zeros and extrema of functions as applied to search and recognition // Cybernetics and Systems Analysis. 2001. Vol. 37. Is. 4.

* MIMD (англ. Multiple Instruction Stream, Multiple Data Stream — множественный поток команд, множественный поток данных) — концепция архитектуры компьютера, используемая для достижения параллелизма вычислений. Машины имеют несколько процессоров, которые функционируют асинхронно и независимо.

НОВОСТИ

«Траектория будущего» подготовила 150 ИТ-тьюторов в помощь московским учителям

Агентство «Траектория будущего» (АТБ) к началу нового учебного года подготовило команду из 150 ИТ-тьюторов, которые помогут московским учителям освоить новые технологии — инструменты для ведения качественного образовательного процесса.

Школьный ИТ-тьютор — это специалист, который помогает участникам образовательного процесса, преимущественно педагогам, адаптироваться к инновационным решениям в их профессиональной деятельности. Школьный ИТ-тьютор проводит консультации по использованию сетевой инфраструктуры, которая включает в себя электронный дневник, электронные учебники и другие функции. Также ИТ-тьютор знакомит преподавателя с возможностями интерактивных панелей, пришедших на смену классическим школьным доскам.

В школьной сетевой инфраструктуре хранится информация об успеваемости учеников и другие данные по учебному процессу. Благодаря консультациям ИТ-тьютора педагог с легкостью заходит в свой виртуальный рабочий кабинет, просматривает активность учеников, ведет отчетность по успеваемости, которая доступна для директора школы, родителя и ученика в режиме онлайн.

После работы с ИТ-тьютором педагог в полной мере использует возможности интерактивной панели, ориентируется в электронной библиотеке, самостоятельно готовит сценарии уроков, превращая занятие в увлекательный процесс и при этом экономя время на разработке темы.

ИТ-тьютор сопровождает педагога в течение такого количества времени, которое необходимо учителю для полноценной адаптации к новым решениям.

«У нашего проекта нет аналогов в России. Мы выявили потребность в консультантах по внедрению инновационных технологий в различные сферы и создали новую профессию — ИТ-тьютор, — сказала генеральный директор АТБ Юлиана Науменко. — Этой профессии не обучают в нашей стране. В университетах только появляются первые программы дополнительного обучения на позицию “тьютор”, в которой совмещены технологии, социология и психология. Но потребность в междисциплинарных специалистах актуальна уже сейчас. Внедрение инновационных технологий проходит массово, затрагивает многие сферы и огромные пласты населения, и для каждой профессии, будь то врач или учитель, нужна адаптация к новым инструментам работы. Пройти адаптацию в кратчайшие сроки, не нарушая рабочего процесса, помогает ИТ-тьютор».

В течение летнего периода 2018 года АТБ рассмотрели заявки 1,5 тыс. претендентов, желающих пройти стажировку на позицию «ИТ-тьютор». Отбор кандидатов проходил в несколько этапов: конкурс, тесты, обучение, аттестация. Среди соискателей отдавалось предпочтение студентам и недавним выпускникам с техническим или педагогическим образованием. Около 150 кандидатов прошли отбор и получили возможность освоить новую профессию на стыке наук.

(По материалам CNews)