УДК: 519.688

ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ СОРТИРОВОК ЦЕЛОЧИСЛЕННЫХ МАССИВОВ

Некрасов А. Н., студент гр. ИИб-231, II курс
Научный руководитель: Назимов А. С., к.т.н., доцент
Кузбасский государственный технический университет
имени Т. Ф. Горбачева, г. Кемерово

Развитие технологий привнесло значительные качественные и количественные изменения в информационной области. Многократно возросли и при этом были значительно упрощены способы получения, хранения, обработки информации и ее обмена, особенно с появлением большого количества недорогих и компактных персональных компьютеров и развитием глобальной сети Интернет. Современный уровень технологического прогресса требует умения эффективно работать с информацией, получать из большого количества различных входных данных ключевые аспекты и определять зависимости. В связи с этим большое значение приобретает быстрая и качественная обработка и представление информации.

Одной из наиболее простых и наиболее важных составляющих работы с данными является их сортировка, при этом существует достаточно большое количество алгоритмов, производящих это действие, как существует и достаточное количество готовых фрагментов кода, реализующих различные виды сортировки на разных языках программирования. Вместе с тем, для быстрой сортировки данных в современных языках программирования предусмотрена специальная функция, которая производит сортировку по возрастанию за короткий промежуток времени, однако она теряет свою эффективность в случаях, когда сортировка не ограничивается исключительно полным преобразованием массива данных (например, когда требуется избирательная сортировка или требуется посчитать количество перестановок). В этом случае целесообразно использование иных алгоритмов сортировки, более гибких по сравнению со встроенной функцией.

Целью данной работы является оценка эффективности наиболее распространенных алгоритмов сортировок массивов с числовыми данными.

Важнейшим критерием оценки алгоритма сортировки является время его выполнения. Рассмотрим программную реализацию различных алгоритмов сортировок на языке программирования C#, исполняемых в среде Visual Studio на персональном компьютере (ноутбуке) средней производительности: процессор Intel(R) Core(TM) i3-1005G1 CPU @ 1,20 GHz 1,19 GHz; оперативная память 8,00 Гб (доступно 7,8 Гб); 64-разрядная операционная система, OS Windows 10 PRO (сборка 19045.5487).

В качестве исследуемых алгоритмов выбраны: сортировка пузырьком (Bubble Sort), сортировка вставками (Insertion Sort), сортировка выбором (Selection Sort) и сортировка слиянием (Merge Sort).

Сортировка пузырьком является одной из наиболее простых и часто используемых сортировок, однако предполагает большое количество перестановок. При этом массив разделяется на две части (упорядоченную и неупорядоченную), элементы перемещаются в неупорядоченной части путем последовательных обменов с соседними. На каждой итерации наименьший элемент поднимается на верхнюю границу неупорядоченной части, а размер (глубина) упорядоченной части увеличивается на единицу.

Сортировка вставками заключается в том, что на каждой итерации элемент на границе неупорядоченной части вставляется на нужную позицию в упорядоченной. При сортировке выбором также происходит разделение на упорядоченную и неупорядоченную части, при этом минимальный элемент из неупорядоченной части становится последним в упорядоченной [1].

Сортировка слиянием предполагает разбиение массива на более мелкие отсортированные составляющие, которые затем попарно объединяются циклическим выбором доступных на данный момент элементов из двух массивов [2].

В качестве эталонной сортировки для исследования выберем интроспективную сортировку (комплексную сортировку на основе быстрой и пирамидальной сортировок), реализованную на языке программирования C# в виде встроенного статического метода Array.Sort() [3], [4].

Для проведения эксперимента разработана программа на языке C#, производящая сортировку целочисленных массивов, с использованием компонентного и объектно-ориентированного подходов [4]. В целях чистоты эксперимента все исследуемые алгоритмы сортировки (пузырьком, выбором, вставками и слиянием) содержат исключительно циклические структуры, без использования рекурсивных алгоритмов. Поскольку время сортировки массива можно отчасти отнести к стохастическим процессам, то для точности эксперимента будем соблюдать следующие условия:

- Каждый массив целых чисел генерируется случайным образом; генерация чисел производится в диапазоне [0; N), где N – длина массива.
- Массив случайных целых чисел генерируется 100 раз; после каждой генерации массива над его копиями производится сортировка всеми вышеперечисленными алгоритмами, после чего исходный массив удаляется и производится новая генерация.
- Для каждого алгоритма измеряется время, затраченное на сортировку массива, которое после выполнения алгоритма заносится в специализированный массив и хранится там пока не будут выполнены все генерации массивов одинаковой размерности. В качестве данных, используемых для дальнейшего исследования и сохраняемых отдельно используются лучшее (наименьшее) время из указанного массива и математическое ожидание всех времен в массиве. При увеличении длины обрабатываемого массива массив, хранящий временны́е данные, очищается.

- Длина массива последовательно изменяется с шагом в 100. Минимальное значение элементов в массиве равно 100. Максимальная длина для каждого алгоритма выбирается программой индивидуально, на основании следующих условий: во-первых, должны быть отсортированы массивы, длина которых находится в диапазоне [100; 25000], вне зависимости от времени, затраченного на сортировку. Во-вторых, лучшее время сортировки не должно превышать 50000 мкс. При несоблюдении этих условий сортировка массивов данным алгоритмом прекращается.
- Интроспективная сортировка выполняется над копией массивов до тех пор, пока выполняется хотя бы одна сортировка.

На основании полученных данных для лучших и средних значений времени каждого алгоритма сортировки построены графики зависимости $T_{min} = f(\mathrm{N})$ и $\bar{T} = f(\mathrm{N})$ соответственно, где T_{min} – время сортировки, $\bar{T}$ – среднее время сортировки, N – длина массива (Рисунки 1-2; время указано в мкс). При помощи программных возможностей Microsoft Excel для функций минимального времени определим линии тренда для каждого уравнения сортировки, уравнения функций (степенные функции) и коэффициенты аппроксимации.

Графические отображения данных для сортировок пузырьком, вставками и выбором (Рисунок 1) свидетельствует о том, что вместо уравнений степенной функции возможно рассматривать полиномиальные уравнения второй степени, которые имеют более высокий коэффициент аппроксимации, чем степенные функции (1 для сортировки пузырьком и выбором; 0,9999 для сортировки вставками), однако квадратные уравнения функций предполагают отрицательное время сортировки для массивов малых размеров, что невозможно. Аналогично невозможно рассмотрение линейных уравнений функций для интроспективной сортировки и сортировки слиянием.

На основании полученных в ходе эксперимента и построения графиков данных составим таблицу (Таблица 1, все значения указаны в мкс), отражающую временные затраты различных видов сортировки на обработку массивов различной длины. Наличие в таблице минимальных значений времени сортировки при отсутствии средних значений означает, что значение было определено на основании полученных степенных уравнений, а не на основании эксперимента.

При небольших длинах массивов (до 200) эффективность сортировки слиянием значительно уступает времени сортировки выбором и вставками, что, с учетом сложности написания алгоритма «с нуля» без использования рекурсии делает ее неэффективной для малых размеров. При увеличении длины массива эффективность сортировки слиянием заметно возрастает, что позволяет применять ее для обработки больших объемов данных (так, на обработку массива длиной 150000 в среднем уходит менее 0,05 с. Наименее эффективным алгоритмом сортировки является сортировка пузырьком, показывающая высокие значения времени для различных длин массива (так массив длиной 25000 будет обрабатываться около 2 с), что, с учетом максимальной

простоты написания программы позволяет применять его для массивов малого объема. Средние результаты показывают сортировка вставками и выбором, причем при увеличении длины массива эффективность последней возрастает.

Результаты, полученные при экстраполяции степенных уравнений минимального времени различных видов сортировки на большие объемы данных (10^6 элементов и выше), показывают, что использование пузырьковой сортировки, сортировки слиянием и выбором крайне неэффективно при высоких объемах данных (так, сортировка указанными методами массива объемом 10 миллионов займет не менее 14 часов; для сравнения сортировка слиянием такого же объема данных займет 3,5 секунды).

Наиболее быстрым видом сортировки, даже при объеме, равном 10^9, остается интроспективная сортировка. При этом необходимо отметить важную деталь: при увеличении объемов данных эффективность данной сортировки начинает немного снижаться по сравнению с сортировкой слиянием. Так, при малых объемах данных время сортировки слиянием превышало время интроспективной сортировки в 5-10 раз, по достижении значения в миллиард данных интроспективная сортировка оказалась эффективнее лишь в 3 раза. Таким образом, сортировка слиянием может стать заменой интроспективной сортировки, компенсируя скорость выполнения возможностью добавления дополнительных условий сортировки.

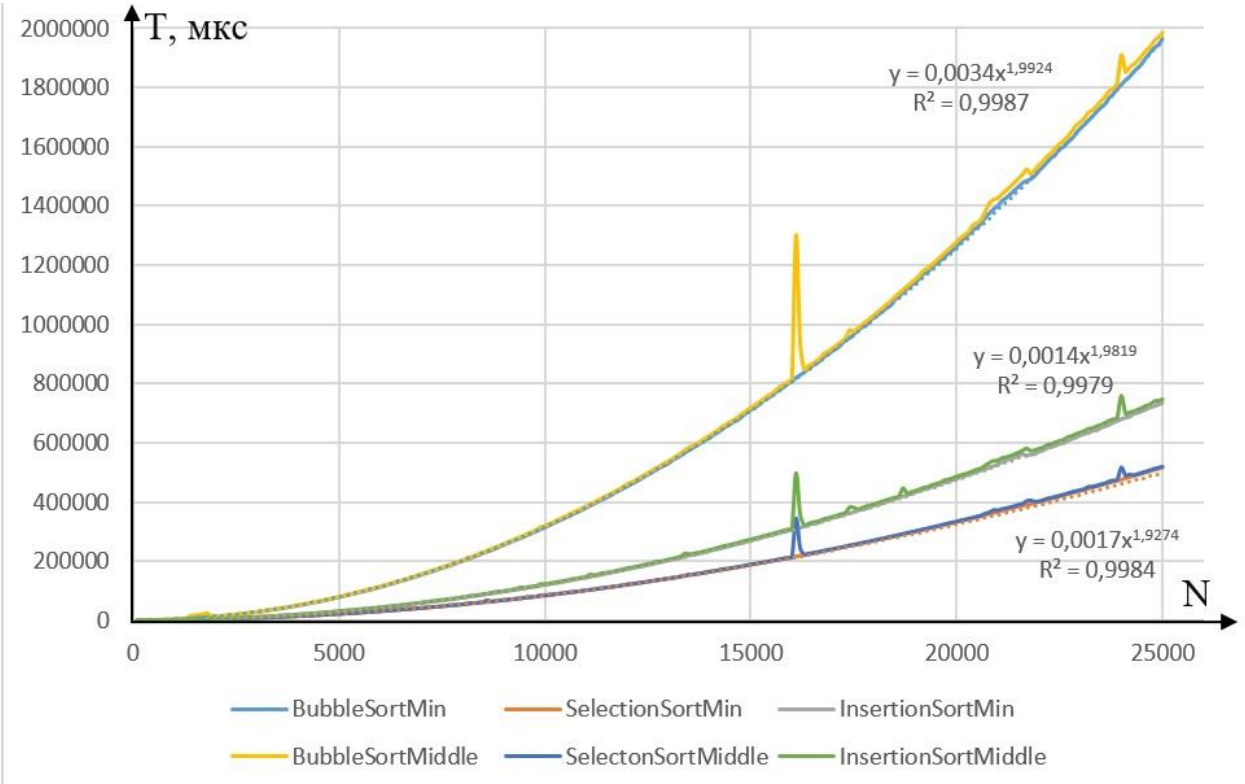

Рисунок 1. Результаты работы алгоритмов сортировок (пузырьком, вставками и выбором)

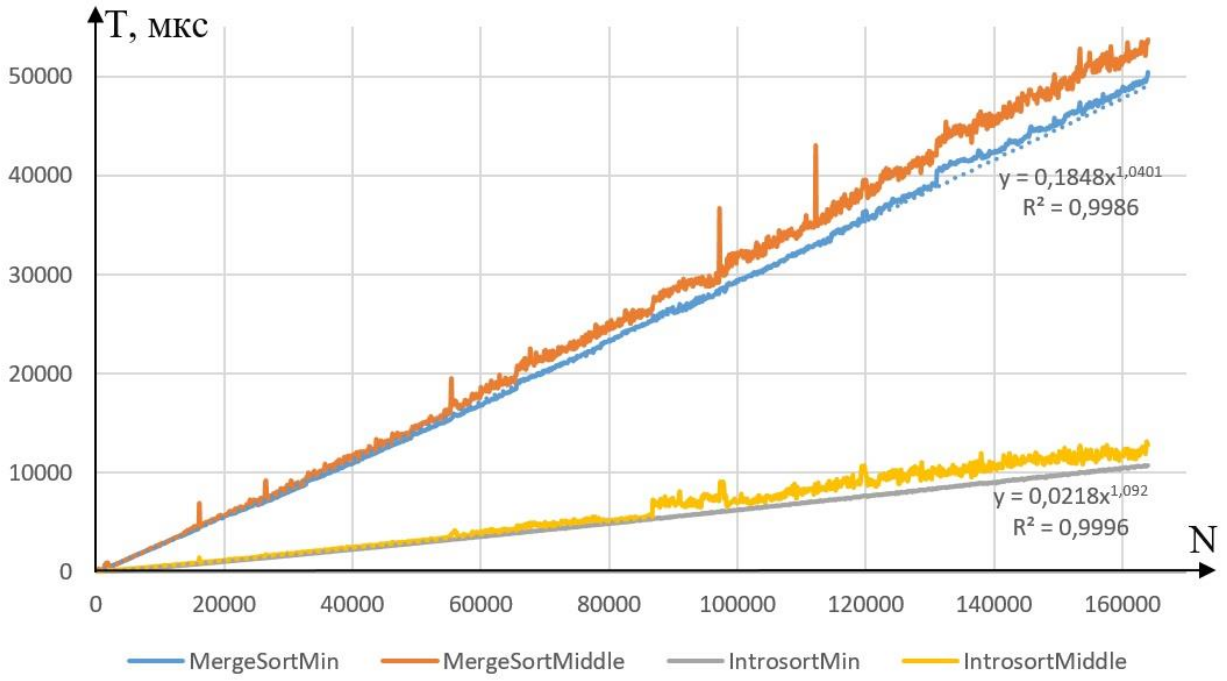

Рисунок 2. Результаты работы алгоритмов сортировок (слиянием и интроспективная сортировка)

Таблица 1.

Минимальное и среднее время сортировки массивов различной длины

Массивы малого объема						
Алгоритм/ размер массива	**100**	**200**	**500**	**1000**	**2500**	**Мин/ среднее**
Bubble Sort	30,6	118,6	773,8	3056,4	19174,8	Мин.
	65,423	179,342	1102,617	3404,14	19694,21	Среднее
Selection Sort	17,1	52,1	267,3	941,5	5429,9	Мин.
	27,118	80,926	402,062	1067,565	5566,009	Среднее
Insertion Sort	13	48,2	291,3	1092	7342,3	Мин.
	43,81	90,005	486,398	1318,887	7687,984	Среднее
Merge Sort	30,8	62,7	104,9	230,3	604,5	Мин.
	134,937	126,983	191,981	274,449	661,132	Среднее
Introsort	2,5	5,8	17,1	38,2	108,9	Мин.
	11,034	7,822	29,458	43,195	114,396	Среднее
Массивы небольшого объема						
	5000	**10000**	**15000**	**20000**	**25000**	
Bubble Sort	77699,6	312770,6	710410,1	1262822,1	1964134,8	Мин.
	79224	318096,9	718506,7	1278397,7	1984034,8	Среднее
Selection Sort	21098,2	83432,1	187387	332025,8	515946	Мин.
	21363,13	83808,22	189216,7	333577,2	518984,5	Среднее
Insertion Sort	29618	119713,8	267677,1	478295,4	734751,2	Мин.
	30494,6	121657,5	273400,7	485325,4	746665,8	Среднее

Merge Sort	1266,1	2638,7	4071,7	5579,8	6904,6	Мин.
	1372,6	2819,2	4475,4	5740,4	7258,3	Среднее
Introsort	242,2	515,1	799,4	1086,3	1401,2	Мин.
	250,713	533,092	823,911	1124,75	1451,59	Среднее
Массивы среднего размера						
	50000	**100000**	**150000**	**200000**	**250000**	
Bubble Sort	7829008	31151497	69875212	123951298	193345732	Мин.
	-	-	-	-	-	Среднее
Selection Sort	1937509	7369685	16100791	28032003	43096152	Мин.
	-	-	-	-	-	Среднее
Insertion Sort	2877509	11366533	25387697	44899281	69872348	Мин.
	-	-	-	-	-	Среднее
Merge Sort	13928,6	29265,2	45221,3	60298	76050	Мин.
	14617,7	31883,4	47828	-	-	Среднее
Introsort	2929,4	6277,6	9744,8	13402	17100	Мин.
	3112,4	6901,8	11131,9	-	-	Среднее
Массивы большого размера						
	500000	**10^6**	**10^7**	**10^8**	**10^9**	
Bubble Sort	$7{,}69\cdot10^{8}$	$3{,}06\cdot10^{9}$	$3{,}01\cdot10^{11}$	$2{,}96\cdot10^{13}$	$2{,}9\cdot10^{15}$	Мин.
Selection Sort	$1{,}64\cdot10^{8}$	$6{,}24\cdot10^{8}$	$5{,}28\cdot10^{10}$	$4{,}46\cdot10^{12}$	$3{,}78\cdot10^{14}$	Мин.
Insertion Sort	$2{,}76\cdot10^{8}$	$1{,}09\cdot10^{9}$	$1{,}05\cdot10^{11}$	$1\cdot10^{13}$	$9{,}62\cdot10^{14}$	Мин.
Merge Sort	156387	321590	3526972	38681381	424230576	Мин.
Introsort	36453	77706	960410	11870158	146708911	Мин.

Для оценки эффективности алгоритмов сортировок в качестве критерия наряду с временем выполнения может выступать разброс между средним и минимальным значениями. Для оценки относительного расхождения между средним и минимальным значениями воспользуемся формулой:

$$\varepsilon = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{\overline{T_i}-T_{\min(i)}}{\overline{T_i}}\right)\cdot 100\% \qquad (1)$$

где:

$\overline{T_i}$ [мкс] – среднее время сортировки массива заданной длины;

$T_{\min(i)}$ [мкс] – минимальное время сортировки массива заданной длины;

n – количество отсортированных массивов различной длины;

ε [%] – среднее относительное расхождение между средним и минимальным временем сортировки.

Результаты расчетов приведены в Таблице 1 (значения округлены до тысячных).

Таблица 2.

Среднее относительное расхождение между средним и минимальным временем сортировки

Алгоритм	**ε, %**
Bubble Sort	3,648

Selection Sort	3,167
Insertion Sort	4,889
Merge Sort	6,696
Introsort	10,598

Наименьшие значения расхождения демонстрируют сортировка выбором, пузырьком и вставками; наибольшее расхождение имеет интроспективная сортировка. Сопоставляя время выполнения и относительное расхождение можно констатировать, что функции, имеющие большое время выполнения имеют меньшее относительное расхождение и наоборот. Однако, учитывая кратное превышение времени выполнения сортировок пузырьком, вставками и слиянием, большее значение относительного расхождения на уровень эффективности сортировки слиянием и интроспективной сортировки не влияет.

Полученные результаты позволяют сделать следующие выводы:

1. Для выполнения задачи простой полной сортировки массива любой длины наиболее удобна интроспективная сортировка, имеющая многократное превосходство по времени.
2. Для обработки массивов малых и средних размеров наиболее удобна сортировка выбором, которая показывает небольшое относительное расхождение между средним и минимальным временами сортировки и имеет достаточно простой алгоритм.
3. При обработке малых массивов возможно использовать пузырьковую сортировку, недостатком которой является большое время выполнения алгоритма, а преимуществом – простота его написания.
4. Для обработки больших массивов данных с возможностью добавления условий рекомендуется использовать сортировку слиянием, имеющую достаточно малое время выполнения, при этом крайне не рекомендуется использование сортировок выбором, вставками и пузырьком, требующих очень больших временных затрат.

Список источников

1. Ландовский В.В. Алгоритмы обработки данных: учеб. пособие. / В.В. Ландовский – Новосибирск: Изд-во НГТУ, 2018. – 65 с.

2. Array.Sort Метод // Microsoft Learn. [Электронный ресурс] // Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/api/system.array.sort?view=net-8.0 (дата обращения 09.03.2025)

3. Introsort // РУВИКИ. [Электронный ресурс] // Режим доступа: https://ru.ruwiki.ru/?curid=2831532&oldid=1175531563 (дата обращения: 09.03.2025).

4. Пространство имен Microsoft.Office.Interop.Excel [Электронный ресурс] // Режим доступа: https://learn.microsoft.com/ru-

ru/dotnet/api/microsoft.office.interop.excel?view=excel-pia (Дата обращения 21.02.2023)