

## Анализ сортировок одномерных массивов.

Абрамова Елизавета Ильинична

11 класс, МБОУ «Водоватовская СШ».

Научный руководитель: С.И. Галкин.



Данная исследовательская работа проведена в целях определения эффективного способа обработки информации. Существует множество алгоритмов, позволяющих сортировать набор данных в определенном порядке, но мало какие экономят наше время. В среде программистов пользуется авторитетом сортировка Хоара (быстрая сортировка). В своей работе я разберу шесть более популярных алгоритмов, определю самую эффективную из них, далее сравню показатели с сортировкой Хоара.

Я реализовала шесть алгоритмов сортировки на одном из самых быстрых языков программирования – C++.

Для удобства создала небольшое меню (Рис.1), которое позволяет:

- Задавать длину массива
- Выводить массив на экран
- Реализовывать сортировку массива
- Выводить на экран время, затраченное на работу алгоритма и количество совершенных им действий.

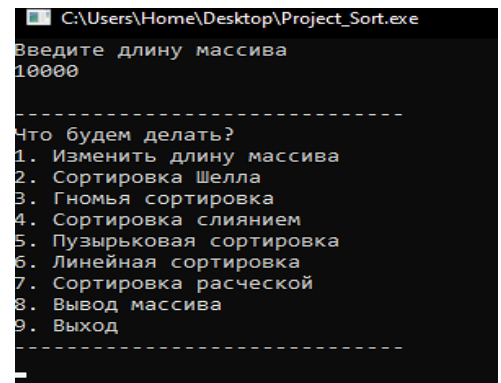


Рис.1 Скриншот меню.

Для исследования были взяты следующие алгоритмы:

- Сортировка Шелла
- Гномья сортировка
- Сортировка слиянием
- Пузырьковая сортировка
- Линейная сортировка
- Сортировка расческой

Для каждого алгоритма массив заполнялся случайными значениями. Время работы, для точности вычислений, было решено представить в секундах.

Данные, полученные в ходе работы программы были занесены в таблицу и там же рассчитано:

1. Среднее время перестановки (Рис.2, 5 столбец.)
2. Среднее количество перестановок (Рис.2, 6 столбец)

сортировки	кол-во элементов	кол-во перестановок	время (мс)	пер. на 1 элем.	время на 1 эл. (мкс)
Сортировка Шелла	100 000	390 872 698	4 603	3 908,7	46,03
Гномья сортировка	100 000	2 498 210 127	34 025	24 982,1	340,25
Сортировка слиянием	100 000	199 999	7 179	2,0	71,79
Сортировка пузырьком	100 000	2 500 109 236	111 918	25 001,1	1 119,18
Линейная сортировка	100 000	747 734	18 713	7,5	187,13
Сортировка расческой	100 000	291 062	132	2,9	1,32

Рис.2 Таблица с вычислениями.

Для наглядности визуализируем полученные данные с помощью гистограмм.

## 1. Среднее значение перестановок. (Рис.3)

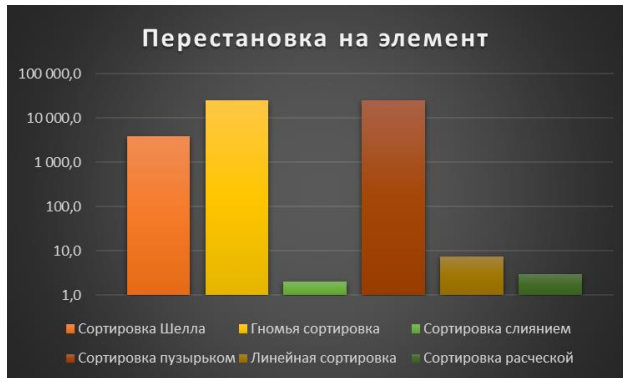


Рис.3 Гистограмма со средним значением перестановок

Сортировки **слиянием**, **расческой** и **линейная** имеют **минимальные** показатели. На первый взгляд, это должно положительно сказаться, на их времени работы. Оно должно стать меньше. В то время, как **гномья**, **пузырьком** и **сортировка Шелла** имеют большой показатель перестановок на элемент. Отметим, что сортировка **слиянием** делает самое минимальное количество перестановок на элемент, кажется, она должна тратить **меньше** всего времени.

Перейдем к диаграмме, на которой отражено время, потраченное на один элемент. Это то, что как раз нас и интересует:

## 2. Среднее время (Рис.4)

Как и ожидалось, алгоритмы, у которых число перестановок на элемент **больше**, соответственно тратят **больше времени**, это **пузырьковая** и **гномья сортировки**. Так же и **линейная сортировка**, которая совершает **мало перестановок**, оказалась совсем **не эффективной**. **Сортировка Шелла**, наоборот предпочитает много времени элементам не уделять, и по времени оказалась даже быстрее нашего первого лидера. При этом сортировка **слиянием** не оправдала наши надежды и уступила **первое место сортировке расческой**.



Рис.4 Гистограмма со средним временем

Теперь выберем сортировку, которая тратит меньше всего времени на один элемент.

**Наш победитель – сортировка расчёской.**

Сортировка Хоара пользуется особой популярностью в среде разработки. Ей дали говорящее название – быстрая сортировка. У данной сортировки есть особые привилегии: во многих средах разработки алгоритм Хоара используется во встроенных функциях сортировки. Разработчики прописали его, чтобы не тратить время там, где его можно сэкономить.

Действительно ли она настолько эффективная?

Сравним показатели сортировки **расчёской** и **быстрой** сортировки.

Чтобы заранее не знать результат, я написала отдельно программу для сортировки **Хоара**. (Рис.5)

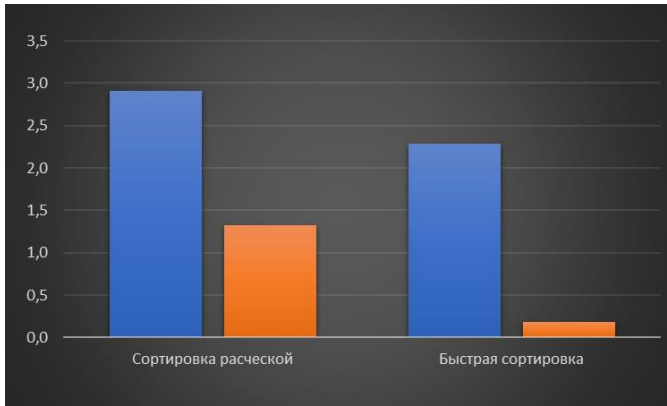
Расчеты велись на массиве той же размерности. Внесем полученные данные в таблицу и рассчитаем те же характеристики.

```
C:\Users\Home\Downloads\Quick_Sort (1).exe
input size of array :
100000
Number swap = 228784
Operating time = 18
Для продолжения нажмите любую клавишу . . .
```

Рис.5 программа для сортировки Хоара

Для наглядности представим полученные результаты в виде гистограммы.

Здесь, **синий столбец** отвечает за количество **перестановок** на элемент, а **оранжевый** отвечает за **время**.



Обратим внимание, как сильно расходятся показатели у этих алгоритмов.

Теперь совершенно каждый может увидеть, что в этом поединке лидирует алгоритм с говорящим названием – **быстрая сортировка** – она написана английским ученым сэром Чарльзом Ёнтони Рйчардом Хёаром в 1960 году, во время его работы в МГУ.

**Рис.6** Сравнительная гистограмма

Таким образом, мы с вами наглядно убедились, что быстрая сортировка, которой больше 60-ти лет, полностью оправдывает своё название и более того, является актуальной и востребованной и по сей день.

#### Литература.

1. *С. В. Глушаков, А. В. Коваль, С. В. Смирнов.* Язык программирования C++. 2001.
2. *Брайан Керниган, Деннис Ритчи.* Язык программирования C. 1978.
3. Описание алгоритмов сортировки и сравнение их производительности [электронный ресурс] URL: <https://habr.com/ru/post/335920/>
4. Алгоритм сортировки [электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Алгоритм\\_сортировки](https://ru.wikipedia.org/wiki/Алгоритм_сортировки)