Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«** **Сортировки на С»**

Выполнил:

студент/ка группы 38200-1

Исаев В.И.

Проверил:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2019

**Содержание.**

Постановка задачи ..................................................................... 3

Метод решения .......................................................................... 4

Руководство пользователя ........................................................ 5

Описание программной реализации ........................................ 6

Подтверждение корректности .................................................. 7

Результаты экспериментов ....................................................... 11

Заключение ................................................................................ 12

**Постановка задачи.**

Цель моей лабораторной работы состояла в исследовании алгоритмов сортировки массивов данных типа *FLOAT.* А также проверка их корректности в зависимости от различных значений.

В данной работе я исследую следующие методы сортировки данных:

1)Сортировка выбором (О(n2) - среднее время)

2)Быстрая сортировка (O(n log n) - среднее время)

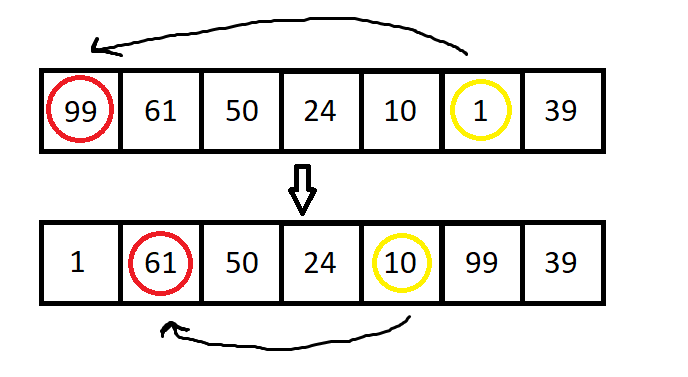
3)Сортировка слиянием (O(n log n) - среднее время)

4)Поразрядная сортировка (O(m \* (n + k)) - среднее время)

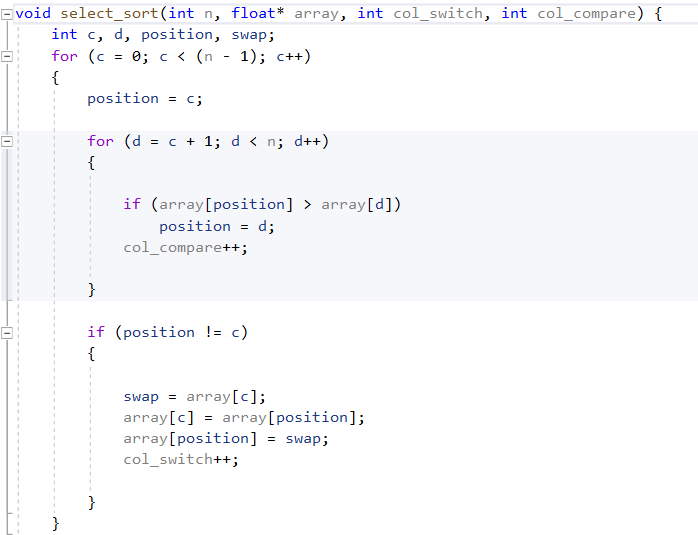
**Метод решения.**

1. ***Сортировка выбором***

Сортировка выбором - здесь, чтобы отсортировать массив, находим элемент с минимальным значением, затем сравниваем его со значением первой неотсортированной позиции. Если этот элемент меньше, то он становится новым минимумом и их позиции меняются.



Код на С:

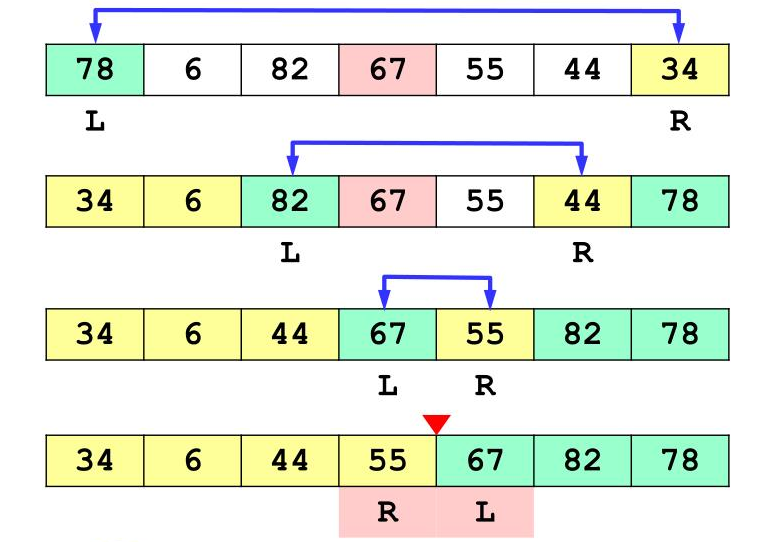
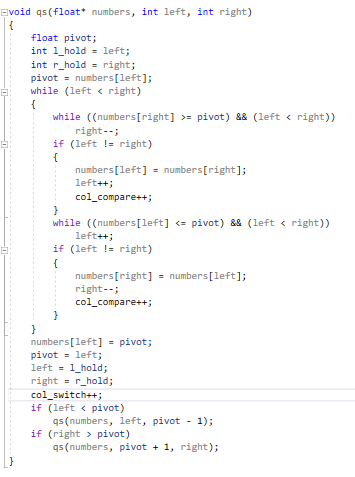


1. ***Быстрая сортировка***

Отличительной особенностью быстрой сортировки является операция разбиения массива на две части относительно опорного элемента. Например, если последовательность требуется упорядочить по возрастанию, то в левую часть будут помещены все элементы, значения которых меньше значения опорного элемента, а в правую элементы, чьи значения больше или равны опорному. Вне зависимости от того, какой элемент выбран в качестве опорного, массив будет отсортирован, но все же наиболее удачным считается ситуация, когда по обеим сторонам от опорного элемента оказывается примерно равное количество элементов. Если длина какой-то из получившихся в результате разбиения частей превышает один элемент, то для нее нужно рекурсивно выполнить упорядочивание, т. е. повторно запустить алгоритм на каждом из отрезков.

Алгоритм быстрой сортировки включает в себя два основных этапа:

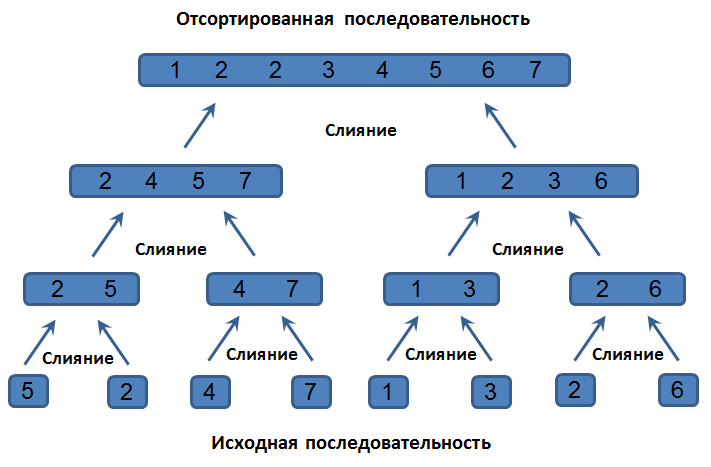
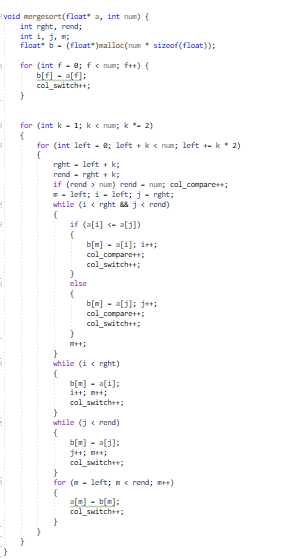
* разбиение массива относительно опорного элемента;
* рекурсивная сортировка каждой части массива.

Код на С:

1. ***Сортировка слиянием.***

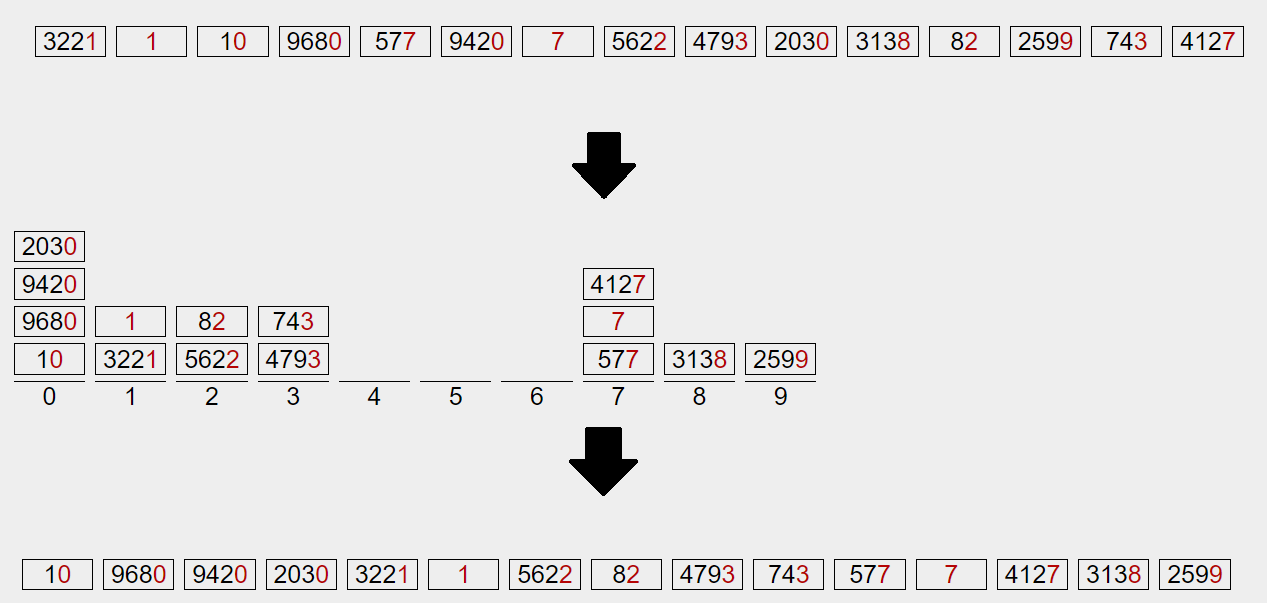
Сортировка слиянием - алгоритм сортировки, который упорядочивает списки в определённом порядке. Сначала задача разбивается на несколько подзадач меньшего размера. Затем эти задачи решаются с помощью рекурсивного вызова или непосредственно, если их размер достаточно мал. Наконец, их решения комбинируются, и получается решение исходной задачи.

1. массив рекурсивно разбивается пополам, и каждая из половин делиться до тех пор, пока размер очередного подмассива не станет равным единице;
2. далее выполняется операция алгоритма, называемая слиянием. Два единичных массива сливаются в общий результирующий массив, при этом из каждого выбирается меньший элемент (сортировка по возрастанию) и записывается в свободную левую ячейку результирующего массива. После чего из двух результирующих массивов собирается третий общий отсортированный массив, и так далее. В случае если один из массивов закончиться, элементы другого дописываются в собираемый массив;
3. в конце операции слияния, элементы перезаписываются из результирующего массива в исходный.

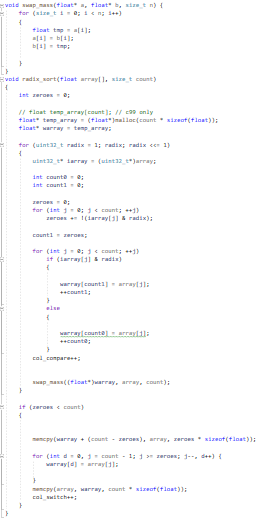
 

1. ***Поразрядная сортировка***

Числа сортируются по разрядам. Существует два варианта: least significant digit (LSD) и most significant digit (MSD). При LSD сортировке, сначала сортируются младшие разряды, затем старшие. При MSD сортировке все наоборот. При LSD сортировке получается следующий порядок: короткие ключи идут раньше длинных, ключи одного размера сортируются по алфавиту, это совпадает с нормальным представлением чисел: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.



Код на С:



**Руководство пользователя.**

Опишем руководство по пользованию моей программой.

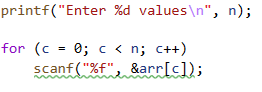
1)Первое, что встречает пользователя при запуске - просьба ввести количество элементов в массиве.



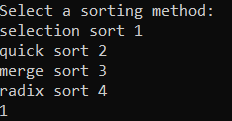


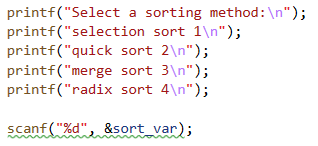
2)После введения количества элементов, программа попросит ввести нас непосредственно сами элементы.



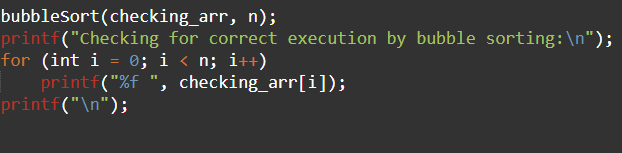
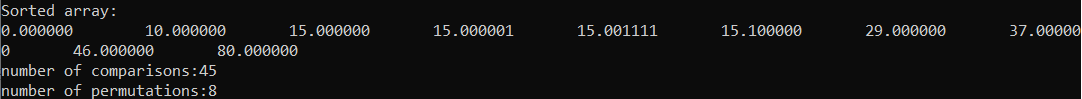


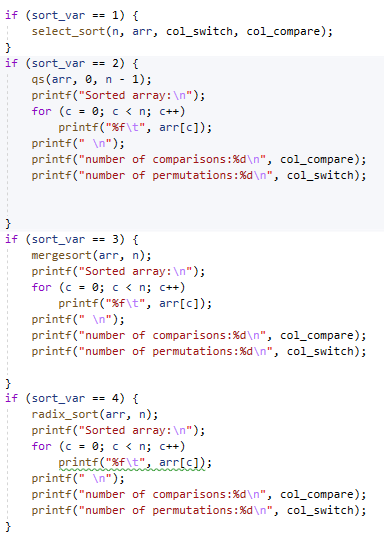
3)Затем программа запрашивает пользователя выбрать метод сортирования данных.





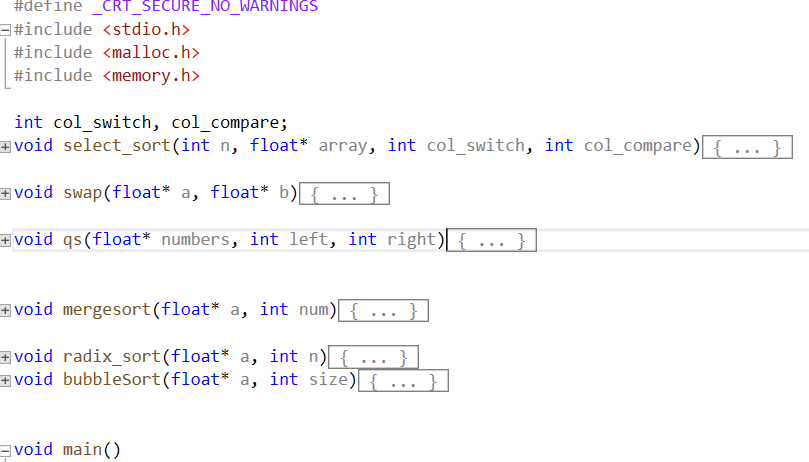
4)После выбора сортировки происходит непосредственно сама сортировка данных. На экран выводится отсортированный массив и посчитанное количество перестановок и сравнений, которые были сделаны за время выполнения алгоритма. А также выводится массив, отсортированный методом пузырька, для проверки корректности выполнения сортировки.





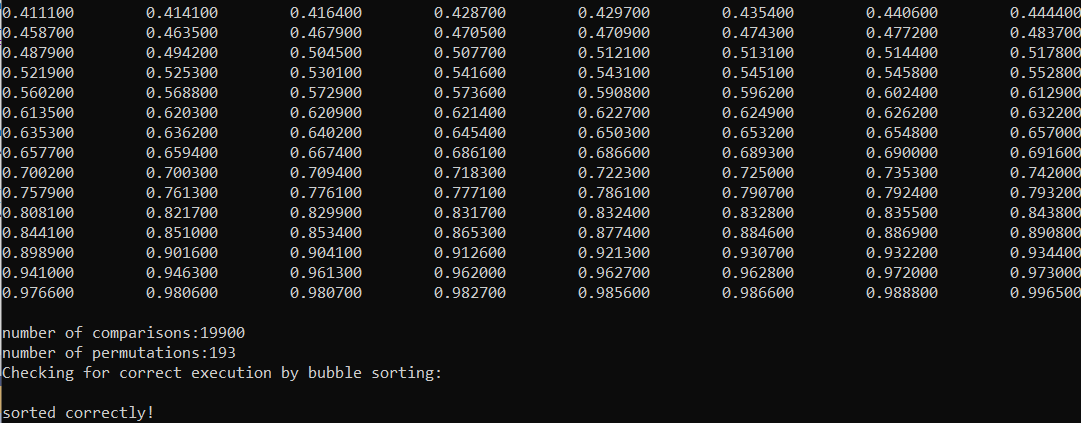
**Описание программной реализации.**

Данная программа выполнена монолитно, в одном файле содержатся все компоненты программы. Сортировки выполнены в виде вызываемых функций, принимающие аргументы в виде сортируемого массива. После вывода отсортированного массива осуществляется сортировка пузырьком для проверки корректности выполнения сортировки методом, выбранном ранее.

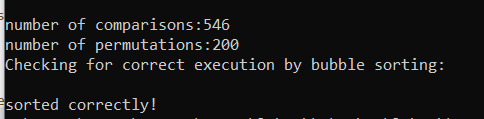


**Результаты экспериментов.**

Сгенерируем разное количество случайных чисел типа *FLOAT.*

Сортировка вставками корректно завершила свою работу для 200 чисел.

Такой же результат для быстрой сортировки



Аналогичные результаты были получены для поразрядной сортировки и сортировки слиянием.

Из чего мы можем сделать вывод о корректности выполнения данных сортировок.

**Заключение.**

В заключение стоит отметить, что в ходе работы итоговая цель была достигнута путем рассмотрения разных вариантов реализации алгоритмов сортировки данных и выбора из них наиболее понятных и простых. Однако нужно сделать важное замечание, что в ходе работы программа не один раз дорабатывалась и подвергалась разного рода тестированиям(увеличение количества данных, увеличение самих входных данных(в пределах допустимых значений типа float))