Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Нижегородский государственный университет им. Н.И. Лобачевского»

**Институт информационных технологий, математики и механики**

**Отчет**

по лабораторной работе №3

**«Поразрядная сортировка для целых чисел со слиянием «разделяй и властвуй»»**

**Выполнил:**

студент группы 381608

Киселев Д. С.

**Проверил:**

Нижний Новгород

2018

**Содержание**

[Постановка задачи 2](#_Toc531367741)

[Метод решения 3](#_Toc531367742)

[Схема алгоритма 4](#_Toc531367743)

[Описание программной реализации 5](#_Toc531367744)

[Подтверждение корректности 6](#_Toc531367745)

[Результаты экспериментов 7](#_Toc531367746)

[Заключение 8](#_Toc531367747)

[Приложение 9](#_Toc531367748)

# Постановка задачи

**Цель** – повышение эффективности поразрядной сортировки целых чисел при помощи распараллеливания алгоритма, основанного на сортировке слиянием «Разделяй и властвуй».

Входные данные – массив случайных целых чисел. Выходные данные – отсортированный массив целых чисел.

**Критерий эффективности разработки** – сокращение времени, затрачиваемого на сортировку массива данных.

**Требуется**:

1. Проанализировать последовательный алгоритм поразрядной сортировки целых чисел слиянием.
2. Найти часть алгоритма, которую можно распараллелить.
3. Реализовать этот параллельный алгоритм на языке Си, используя технологию MPI, и протестировать.

# Метод решения

Принцип работы сортировки слиянием «разделяй и властвуй»: задача разбивается на подзадачи меньшего размера, которые решаются по отдельности, после чего их решения комбинируются для получения решения исходной задачи.

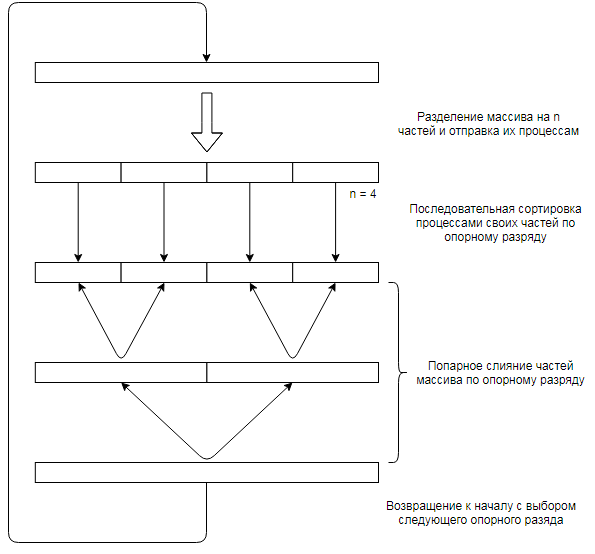
Из самого принципа работы ясно, что для эффективного распараллеливания этого алгоритма нужно распараллеливать ту часть, в которой задача разбивается на подзадачи меньшего размера.

При этом учтем, что для того, чтобы сортировать наши числа поразрядно, нам нужно будет отсортировать эти числа столько раз, сколько в этих числах содержится разрядов. Будем начинать сортировку в сторону старших разрядов от младших.

В итоге получился следующий метод решения:

1. Берем самый младший разряд чисел за опорный.
2. Разбиваем исходный массив случайных чисел на n (кол-во процессов) примерно равных частей.
3. Передаем части массива соответствующим процессам.
4. Каждый процесс при помощи любой последовательной сортировки сортирует свою часть массива целых чисел по опорному разряду.
5. Затем все процессы разбиваются на пары и проводят процесс слияния своих частей массива (по опорному разряду). Процессы, которые в итоге получили слитые части снова разбиваются на пары и снова сливают свои части. Этот процесс завершается, когда последняя пара процессов, которые имеют примерно равные половинки массива, сливают свои половинки в полный массив.
6. Если это не последний обрабатываемый разряд чисел, то возвращаемся в начало алгоритма и берем за опорный следующий разряд чисел, иначе завершаем работу.

# Схема алгоритма



# Описание программной реализации

**Руководство пользователя**

Запуск программы в консоли ОС:

*>mpiexec –n <proc\_num> lab3.exe <size> <lower value> <upper value>*

Где:

* proc\_num – количество процессов
* size – размер исходного массива случайных чисел
* lower value – нижняя граница генерации случайных целых чисел
* upper value – верхняя граница генерации случайных целых чисел

**Руководство программиста**

Код программы можно просмотреть в разделе «[Приложение](#_Приложение)».

# Подтверждение корректности

Для подтверждения корректности в программе используется функция qsort() из библиотеки «stdlib.h».

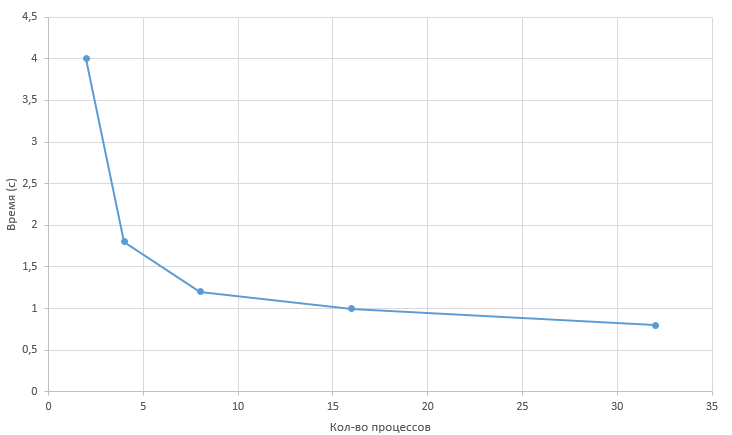
Массив, отсортированный нашим алгоритмом, последовательно сравнивается с массивом, отсортированным функцией qsort().

Так же для наглядности в консоль выводится исходный массив и отсортированный, но только при условии, что его размер не превышает 32 элемента.

# Результаты экспериментов

По данным экспериментов видно, что реализованный алгоритм успешно справляется со своей задачей.

Эффективность работы можно оценить на графике ниже:



Тестирование эффективности проводилось на массиве целых чисел размером в 50000 элементов при интервале генерации в пределах [-10000, 10000].

# Заключение

Поставленная задача была успешно выполнена. Был реализован и протестирован алгоритм поразрядной сортировки целых чисел при помощи слияния «разделяй и властвуй».

# Приложение

Исходный код программы можно найти по адресу:

<https://github.com/your0mind/parallel-programming/blob/kiselev_ds_task3/1608/kiselev_ds/task3/main.c>