

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

СОРТИРОВКА

Отчет по лабораторной работе №6 по дисциплине
«структуры и алгоритмы обработки данных в ЭВМ»

Обучающийся гр. 439-3
(группа)

А. С. Мазовец
(подпись) (И. О. Фамилия)

« _____ » _____ 2021 г.
(дата)

Проверил:

ассистент кафедры АСУ
(должность, ученая степень, звание)

Я. В. Яблонский
(подпись) (И. О. Фамилия)

_____ « _____ » _____ 2021 г.
(оценка) (дата)

1 Задание на лабораторную работу

1.1 Вариант 6

Пусть даны три файла целых положительных чисел, состоящих из 100, 1000 и 10000 чисел. Выполнить сортировку данных с помощью метода двоичной быстрой сортировки.

2 Алгоритм решения задачи

- 1) Разделить массив на 2 подмассива по последнему биту;
- 2) Выполнить алгоритм на этих подмассивах по следующему биту;
- 3) После прохода по последнему биту объединить результаты в один массив.

3 Листинг программы

3.1 Модуль сортировочного алгоритма

```

1 | module Sorting where
2 |
3 | import Data.Bits
4 | import Data.List
5 |
6 | quickSortB :: (Num a, FiniteBits a, Ord a) => [a] -> [a]
7 | quickSortB [] = []
8 | quickSortB l @ (x:_) = flip qsortB' l $ finiteBitSize x
9 |   where
10 |     qsortB' _ [] = []
11 |     qsortB' r l' | r == 0 = concat $ divByBit r l'
12 |                  | otherwise = concat . map (qsortB' $ r-1) $
13 |                      divByBit r l'
14 |     divByBit r = sequence [fst, snd] . partition ((==0) . (.&.
15 | setBit 0 r))

```

3.2 Основной модуль

```

1 | import Sorting
2 |
3 | main :: IO ()
4 | main = do
5 |     list <- quickSort
6 |     . map (read :: String -> Int)
7 |     . lines
8 |     <$> readFile "I"

```

```

9 |      return ()
10 |      print $ head list
11 |      print $ last list
12 |      return ()

```

4 Пример решения задачи

Во входных файлах 100, 1000, 10000 содержится соответствующее количество числовых значений (См. Рис. 4.1). Для каждого файла выводится отсортированный, по возрастанию набор значений (См. Рис. 4.2)

```

21 0440024401207
20 8174470731420
19 1974079102607
18 9617792228089
17 7123165558898
16 9490822552610
15 296907051565
14 5168400252382
13 5558042907521
12 1917507896756
11 8448176305726
10 5250215475893
9 3814627385170
8 6463062438750
7 2290861551141
6 2533915767377
5 2439728267398
4 9040643688759
3 7138520651963
2 4881641503434
1 9011466747952
1000000 8956299559938

```

Рисунок 4.1 — Входной файл

```
[420317, 23122916, 29400512, 77297365, 88047940, 96997700, 110369735, 144159352, 153386406, 157400122, 174856196, 176
```

Рисунок 4.2 — Выходной файл

5 Вывод

Был освоен и реализован метод быстрой поразрядной сортировки и отсортированы наборы данных для каждого из трех файлов.