

# Отчет

## Лабораторная работа 5

1. Задача. 5.33 Для каждой строки сформировать новую строку, поместив в нее все слова исходной строки, упорядоченные по длине и по алфавиту. (Использовать структурное представление данных состоящую из символов)
2. Алгоритм. На Рис. 1 изображен алгоритм программы, изображение в лучшем качестве находит под именем Algorithm.txt.



### Рис. 1 Алгоритм

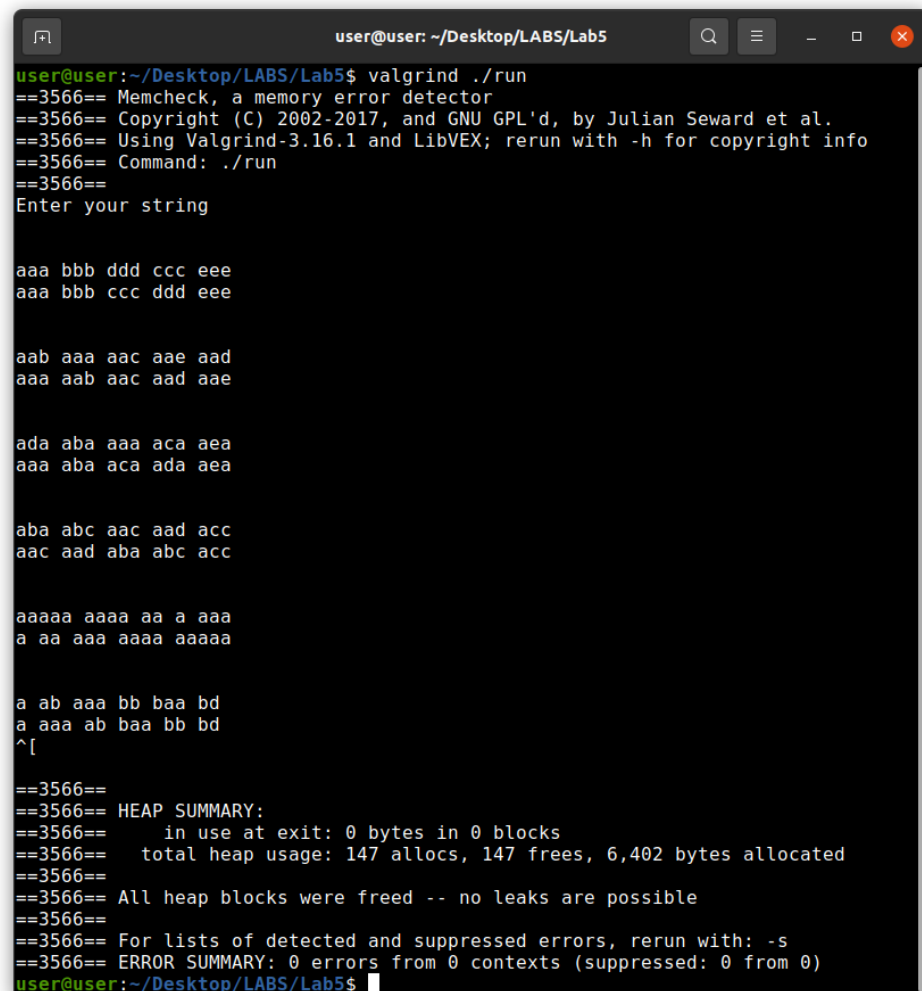
### 3. Тестирование программы. На Рис. 2 приведен пример работы программы.

Входные данные:

- aaa bbb ddd ccc eee (Проверка сортировки слов по первому символу)
- aab aaa aac aae aad (Проверка сортировки слов по последней букве)
- ada aba aaa aca aea (Проверка сортировки по второй букве)
- aba abc aac aad acc (Проверка сортировки по всем буквам кроме первой)
- aaaaa aaaa aa a aaa (Проверка сортировки по длине)
- a ab aaa bb baa bd (Смешенная проверка по длине и алфавитному)

Следует заметить, сортировка в алфавитном порядке имеет больший приоритет чем по длине.

Так же была проведена проверка Valgrind, ошибок не обнаружено.



```
user@user: ~/Desktop/LABS/Lab5
user@user:~/Desktop/LABS/Lab5$ valgrind ./run
==3566== Memcheck, a memory error detector
==3566== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3566== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==3566== Command: ./run
==3566==
Enter your string

aaa bbb ddd ccc eee
aaa bbb ccc ddd eee

aab aaa aac aae aad
aaa aab aac aad aae

ada aba aaa aca aea
aaa aba aca ada aea

aba abc aac aad acc
aac aad aba abc acc

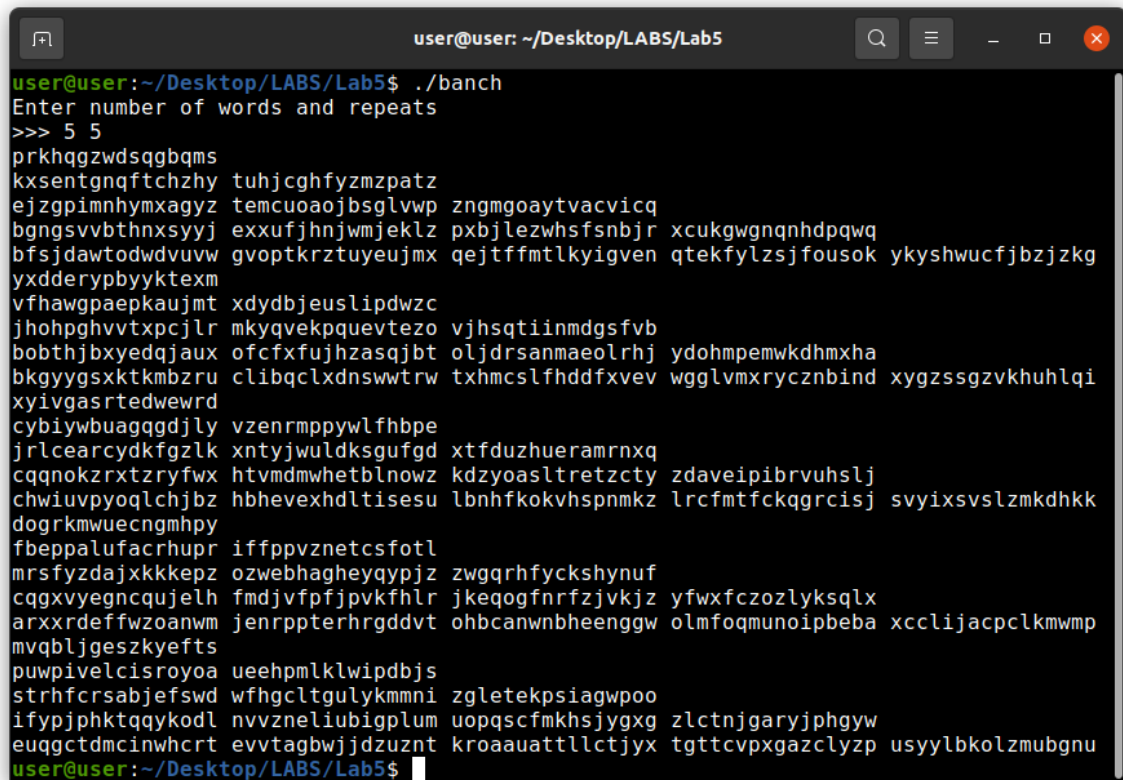
aaaaa aaaa aa a aaa
a aa aaa aaaa aaaaa

a ab aaa bb baa bd
a aaa ab baa bb bd
^[

==3566==
==3566== HEAP SUMMARY:
==3566==    in use at exit: 0 bytes in 0 blocks
==3566==   total heap usage: 147 allocs, 147 frees, 6,402 bytes allocated
==3566==
==3566== All heap blocks were freed -- no leaks are possible
==3566==
==3566== For lists of detected and suppressed errors, rerun with: -s
==3566== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
user@user:~/Desktop/LABS/Lab5$
```

**Рис. 2 Тестирование программы**

4. Генерация. На Рис. 3 приведен вывод работы генерации слов. Результаты тестирования времени сортировки сохраняются в отдельный файл Log.txt. Первое вводимое число пользователем характеризует максимальное количество слов в строке, а второе количество повторов. Программа сортирует сначала строку из одного слова, затем из двух и так далее. Время выполнения сортировки каждой строки записывает в файл.



```
user@user: ~/Desktop/LABS/Lab5
user@user:~/Desktop/LABS/Lab5$ ./branch
Enter number of words and repeats
>>> 5 5
prkhqgzwdsqgbqms
kxsentgnqftchzhy tuhjcghfyzmzpatz
ejzgpimnhymxagyz temcuaoajbsglvwp zngmgoaytvacvicq
bgngsvvbthnxsyyj exxufjhnjwmjeklz pxbjlezwhsfnsbjr xcukgwgngnhdpcwq
bfsjdawtodwdvuvw gvoptkrztuyeuimx qejtffmtlkyigven qtekfylzsjfousok ykyshwucfjbzjzkg
yxdderypbbyktxem
vfhawgpaepkaujmt xdydbjeuslipdwzc
jhohpghvvtxpcjlr mkyqvekpquevtezo vjhsqtiinmdgsfwb
bobthjbxqedqjaux ofcxfujhzasqjbt oljdrsanmaeolrhj ydohmpemwkdhmxa
bkgyygsxktkmbzru clibqclxdnswwtrw txhmcslfhddfxvev wgglvmxrycznbind xygzssgzvkhuhlqi
xyivgasrtedwewrd
cybiywbuagggdjly vzenrmppywlfhbpe
jrlcearcydkfgzlk xntyjwuldksgufgd xtfduzhueramrnqx
cqgnokzrxtzryfwx htvmdmwhetblnowz kdzyoaslretzcty zdaveipibrvuhsly
chwiuvpyoqlchjzb hbhevexhdtisesu lbnhfkokvhspnmkz lrcfmtfckqgrcisj svyixsvslzmkdhkk
dogrkmwuecngmhpv
fbepalufacrhufr iffppvznetcsfotl
mrsfyzdajxkkkep zwebhagheyqypjz zwgqrhfycckshynuf
cggxvyegncqujelh fmdjvfpfjpvkfhlr jkeqogfmrzjvkjz yfwxfczozlyksqlx
arxxrdeffwzoanwm jenrppterhrghdvt ohbcanwnbheengw olmfoqmunoipbeba xcclijacpckmwmp
mvqbljgeszkyefts
puwpivelcisroyoa ueehpmlklwipdbjs
strhfcrsabjefswd wfhgcltgulykmmni zgletekpsiagwpoo
ifypjphktqykodl nvvzneliubigplum uopqscfmkhsjygyx zlctnjgaryjphgyw
euqgctdmcinwhcrt evvtagbwjjdzuznt kroaauattllctjyx tgctcvpxgazclyzp usyylbkolzmubgnu
user@user:~/Desktop/LABS/Lab5$
```

Рис. 3 Генерация

5. График зависимости. На Рис. 4 изображен график зависимости времени обработки строки от количества слов в ней. Линия тренда графика приблизительно совпадает с линейной функцией  $O(X)$ .

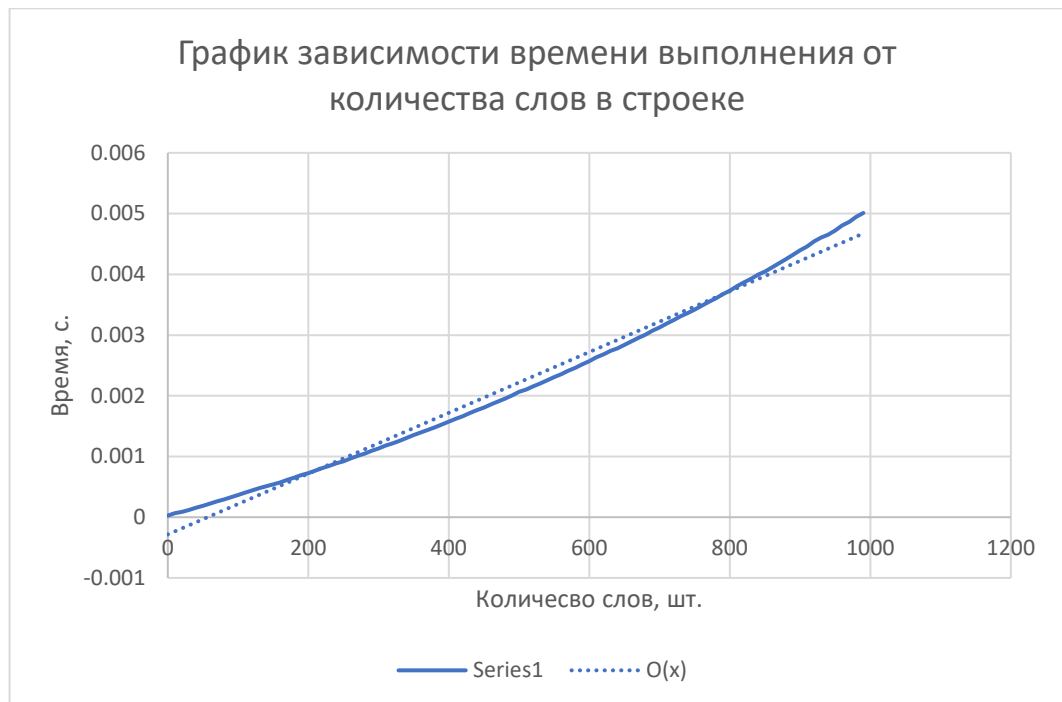


Рис. 4

Вывод: зависимость времени сортировки рассматриваемым методом количества сортируемых данных приблизительно совпала с линейной зависимостью. Однако, можно заметить выпуклость графика, что может говорить о степенной зависимости малого порядка большего единицы, или зависимости  $X \cdot \log(X)$ . Подобный изгиб можно увидеть на графике Рис.5.

Graph for  $\log(x) \cdot x$ ,  $x$

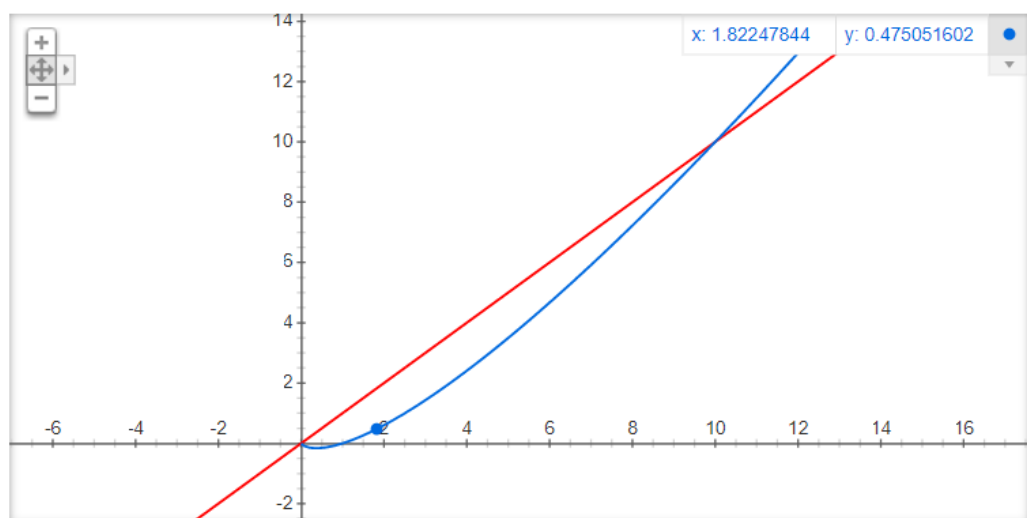


Рис. 5 Пример