

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Дискретный анализ»

Студент: А. В. Барсов
Преподаватель: Н. С. Капралов
Группа: М8О-205Б-19
Дата:
Оценка:
Подпись:

Москва, 2020

Лабораторная работа №1

Задача: Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.

Вариант сортировки: Поразрядная сортировка.

Вариант ключа: телефонные номера, с кодами стран и городов в формате +<код страны> <код города> телефон.

Вариант значения: строки переменной длины (до 2048 символов).

1 Описание

Предполагаем, что каждый из сортируемых элементов массива можно разбить на d разрядов. Тогда, начиная с самого младшего разряда, сортируем массив устойчивым алгоритмом по значению выбранного разряда, и так до самого старшего разряда. Время работы зависит от выбранного устойчивого алгоритма сортировки. Если цифры(разряды) могут принимать значения до небольшого k , то лучший выбор - сортировка подсчётом. Для n чисел с d разрядами значениями от 0 до $k-1$ каждый проход занимает время $\Theta(n + k)$, d проходов - $\Theta(d(n + k))$. Как сказано в [1]: «Если d постоянно и $k = O(n)$, то поразрядная сортировка работает за линейное время».

2 Консоль

```
[artem@IdeaPad solution]$ make
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c main.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c Record.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c Sort.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm main.o Record.o
Sort.o -o solution
[artem@IdeaPad solution$ python generator.py 10 | ./solution
+55-5998-11391278 vvzxRboYGlIpGVCNKpMiZpVzBGzbwPdHDQCdmegJmjgj
+203-6225-54420352 uXlNHoUcNEazrYqHZDQLNlbyoqHFhVevsdkBmHf
+242-6957-77482245 CgaTseqCAWQAilyigyHYJhwvpSsfUlteiTuhxmpRw
+249-8790-13660628 ltKniLmPIWnordqhYvYPlpTXUNh
+476-6224-75326021 uoqlFGghcytKDaqQdWvfNdeQqcFG
+494-2252-79673470 gArtaSpOIfzAtcrSWNRJbbiHVmw
+682-5864-72861170 wZxPLDcQdRYkzQUbSxNymdzZlluAXszPUZsJBBH
+694-5623-51338066 rgGXLzsWxFUdyWVDHsxFiHDjitrlZvgSgWVGfHP
+734-3124-9591930 uzlecHxdnPwtjmXjiYmmHinvAjHuYgpOMruFSKQMfAUZhZlzM
+927-2540-75979719 XzoZudMopyQkvUGxUkrZGpYRvQvbINFEBTo
```

3 Тест производительности

Тест производительности представляет из себя сравнение времени работы поразрядной сортировки с уже существующей устойчивой сортировкой из STL `std::stable_sort`. Тест состоит из 100000 строк ввода.

```
[artem@IdeaPad solution]$ ./wrapper.sh
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c main.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c Record.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c Sort.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm main.o Record.o
Sort.o -o solution
Built main program
g++ -std=c++11 -c benchmark.cpp
g++ -std=c++11 -c ../Record.cpp
g++ -std=c++11 -c ../Sort.cpp
g++ -std=c++11 benchmark.o Record.o Sort.o -o benchmark
Built benchmark
Test generation completed
Count of lines is 100000
Radix sort time: 480216us
STL stable sort time: 543419us
Sorting completed
Comparison completed
rm -f benchmark.o Record.o Sort.o *.txt benchmark
rm -f main.o Record.o Sort.o *.txt solution
```

Как видно, поразрядная сортировка выиграла у обычной устойчивой, но не намного из-за достаточно малого n .

4 Выводы

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я научился эффективно управлять памятью на довольно низком уровне, оптимизировать работу программы путём передачи объектов по адресу, находить и устранять утечки памяти при помощи специальных инструментов. Я понял, как работает реаллокация данных, и что может приводить к утечке памяти, а также к долгому выполнению программы.

Список литературы

- [1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание*. — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))
- [2] *Справочник C++*.
URL: <https://en.cppreference.com/w/> .