Министерство науки и образования РФ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический

университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Пояснительная записка к курсовой работе

на тему:

“Измерение временной сложности алгоритмов”

по дисциплине “Алгоритмы и структуры данных”

Вариант 19

Выполнил студент гр. 4306: Табаков А. В.

Принял: Колинько П. Г.

Санкт-Петербург  
2016

Оглавление

Цель3

1. Задание3
2. Уточнение задания3
3. Способ представления данных в памяти3
4. Последовательность действий3
5. Оценка временной сложности 4
6. График регрессии4

Вывод5

Список используемой литературы6

Приложение7

**Цель**

Экспериментально измерить временную сложность алгоритмов стандартной библиотеки шаблонов языка программирования «C++».

1. **Задание**

Реализовать и исследовать алгоритм выполнения случайной последовательности операций из предоставленных функций:

1. Concat – Сцепление. Вторая последовательность подсоединяется к концу первой, образуя ее продолжение.
2. Erase – Укорачивание. Из последовательности исключается часть с порядковыми номерами из отрезка [p1, p2]
3. Excl – Исключение. Вторая последовательность исключается из первой, если является ее частью.
4. mapAnd – Пересечение. Результатом служит сформированное множество из элементов, которые содержатся и в первом и втором множестве.
5. mapOr – Объединение. Результатом служит сформированное множество из всех элементов, которые содержатся в двух множествах.
6. **Уточнение задания**

Разработать и реализовать в виде программы алгоритм по предложенному индивидуальному заданию. Дать теоретическую оценку временной сложности алгоритма и сравнить её с измерениями. Сделать вывод о временной сложности работы стандартной библиотеки шаблонов языка программирования «C++».

1. **Способ представления данных в памяти**

Данные хранятся в контейнере unordered\_map. Также используется вспомогательный контейнер vector в функциях mapAnd и mapOr для сортировки элементов и последующем использовании функций библиотеки algorithm STL: sort, set\_intersection и set\_union.

Ключ в хеш-таблице unordered\_map произвольное число, а значение номер последовательности.

В каждой функции происходит поиск номера последовательности, что ухудшает временную сложность, но в целом алгоритмы подобраны так, чтобы более-менее оптимизировать процесс.

1. **Последовательность действий**

* concat(B, D)
* concat(C, D)
* mapAnd(B, C, E)
* mapOr(E, C, D)
* excl(C, E)
* excl(B, E)
* erase(E, rand()%20, rand()%100 + 20)
* erase(A, rand() % 20, rand() % 100 + 20)
* concat(C, A)
* concat(C, C)
* mapOr(C, D, A)

1. **Оценка временной сложности**

**Временная сложность алгоритмов STL**

|  |  |  |
| --- | --- | --- |
| Функция | Заявленная средняя временная сложность | Заявленная худшая временная сложность |
| insert | O(1) | O(n) |
| erase | O(1) | O(n) |
| transform | O(n) | O(n) |
| sort | O(n ln n) | O(n ln n) |
| set\_intersection | O(n) | O(n) |
| set\_union | O(n) | O(n) |
| size | O(1) | O(1) |

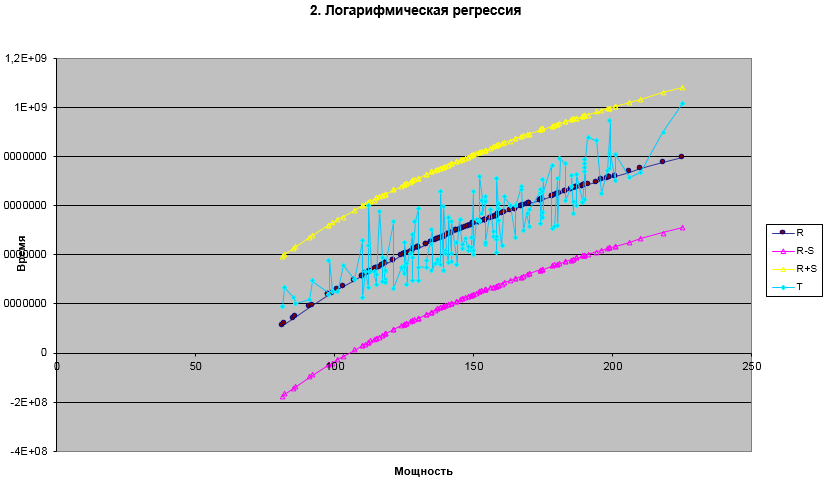
**Временная сложность функций программы**

|  |  |  |
| --- | --- | --- |
| Функция | Средняя | Худшая |
| concat | O(m) | O(m) |
| erase | O(m) | O(n) |
| excl | O(m) | O(n) |
| mapAnd | O(n ln n) | O(n ln n) |
| mapOr | O(n ln n) | O(n ln n) |

прим 1. n – длина первой последовательности, m – длина второй последовательности.

прим 2. mapAnd и mapOr имеют логарифмическую сложность из-за сортировки.

1. **График регрессии**



**Вывод**

Я экспериментально измерил временную сложность алгоритмов стандартной библиотеки шаблонов языка программирования «С++».

Мной была выбрана логарифмическая регрессия, на основании отношений дисперсий. В данной выборке 153 опыта, программа RG32.exe строит уравнения до 6 степени, следовательно, степень свободы выборки была больше чем для 100 опытов (1.39 при 5% погрешности) и меньше чем для 200 опытов (1.26 при 5% погрешности).

Отношения первой дисперсии и остальных больше 3.33, следовательно, сложность не константная, однако последующие отношения меньше 1.16, значит можно сделать вывод, что временная сложность такой последовательности действий логарифмическая.

**Список используемых источников**

1. Алгоритмы и структуры данных: методические указания к лабораторным работам, практическим занятиям и курсовому проектированию. Федеральный образовательный стандарт / сост.: П.Г. Колинько. - СПб.: Изд-во СПБГЭТУ "ЛЭТИ", 2014. - 63 с.
2. Освой С++ самостоятельно за 21 день. Сиддхартха Рао. 688 стр., с ил.; ISBN 978-5-8459-1825-3; 7 издание.
3. http://stackoverflow.com – Сайт вопросов и ответов по программированию.
4. http://cyberforum.ru – Форум программистов и сисадминов.
5. http://www.cplusplus.com/ - Информация о C++

**Приложение**

К данному отчёту прилагаются файлы на электронном носителе:

* Source.cpp – исходный код программы
* in.txt – файл созданный программой, для которого производились расчёты
* out.txt – файл сгенерированный на основе in.txt программой RG32.exe
* TS.xls – электронная таблица, предоставленная Колинько П.Г., с импортированными файлами out.txt и in.txt