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 16

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. Erase – Укорачивание. Из последовательности исключается часть с порядковыми номерами из отрезка [p1, p2]
2. Subst – Включение. Вторая последовательность включается в первую с указанной позиции *p*. Операция похожа на конкатенацию. Сперва берётся начало первой последовательности до позиции *p*, затем идёт вторая последовательность, а за ней — остаток первой.
3. Change – Замена. Вторая последовательность заменяет элементы первой, начиная с заданной позиции *p*.
4. setAnd – Пересечение. Результатом служит сформированное множество из элементов, которые содержатся и в первом и втором множестве.
5. setDif – Различие. Результатом служит сформированное множество из всех элементов, которые содержатся в первом, но не содержится во втором 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 контейнерах: set для хранения множества и vector для поддержки последовательностей в виде вектора итераторов на множество. Для удобства, контейнеры объединены в класс mySet, где values – множество, а pointers вектор итераторов.

1. **Последовательность действий**

setAnd(A, B, E);

setDif(E, A, B);

erase(D, rand() % 300, rand() % 500);

subst(E, B, rand() % 500);

setAnd(C, D, E);

erase(A, rand() % 300, rand() % 500);

setDif(A, C, B);

subst(A, A, rand() % 500);

change(D, A, rand() % 500);

setAnd(D, B, C);

erase(B, rand() % 300, rand() % 500);

subst(D, B, rand() % 500);

setDif(A, E, A);

change(D, A, rand() % 500);

subst(C, A, rand() % 500);

setAnd(B, A, B);

subst(D, B, rand() % 500);

1. **Оценка временной сложности**

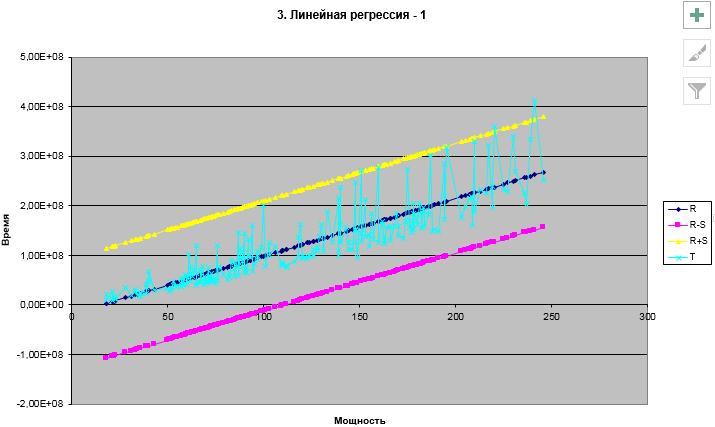
**Временная сложность алгоритмов STL**

|  |  |  |
| --- | --- | --- |
| Функция | Заявленная средняя временная сложность | Заявленная худшая временная сложность |
| vector::push\_back | O(1) | O(n) |
| set::insert | O(ln n) | O(ln n) |
| set\_intersection | O(n) | O(n) |
| set\_difference | O(n) | O(n) |
| swap | O(1) | O(1) |
| size | O(1) | O(1) |

**Временная сложность функций программы**

|  |  |  |
| --- | --- | --- |
| Функция | Средняя | Худшая |
| setAnd | O(n) | O(n) |
| setDif | O(n) | O(n) |
| erase | O(n) | O(n) |
| subst | O(n) | O(n) |
| change | O(n) | O(n) |

1. **График регрессии**



**Вывод**

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

Мною была выбрана линейная регрессия - 1, на основании отношений дисперсий. В нашей выборке 214 опытов, программа RG32.exe строит уравнения до 6 степени, следовательно, степень свободы выборки была больше чем для 200 опытов (1.26 при 5% погрешности).

Отношения первой дисперсии и остальных больше (3.03), следовательно, сложность не константная. Отношения второй дисперсии больше (1.47), при нашем минимуме (1.26), данные отношения нам не подходит. Третьи отношения дисперсий 1.01 – 1.03, и они входят в нужный нам интервал.

На основании представленных данных, можно сделать вывод, что временная сложность такой последовательности действий линейная.

**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.

**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
* V.xls – электронная таблица, предоставленная Колинько П.Г., с импортированными файлами out.txt и in.txt