**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

Кафедра вычислительной техники

Отчет по лабораторной работе №4

по дисциплине «Алгоритмы и структуры данных»

|  |  |
| --- | --- |
| Студент гр. 2307 | Подберёзский А.Д. |
| Преподаватель | Манирагена В. |

Тема: Измерение временной сложности алгоритма в эксперименте на ЭВМ  
Вариант 17

**Содержание**

**Цель работы ................................................................................................. 3**

Задание ........................................................................................................... 3

Описание эксперимента ………………....................................................... 3

Результаты эксперимента ……………………............................................. 3

Таблицы …………………………………………………………………...... 4

**Вывод ............................................................................................................. 6**

Текст программы ........................................................................................... 7

**Цель работы**

## Экспериментальное подтверждение (опровержение) теоретической временной сложности алгоритмов работы с множествами и последовательностями.

## Задание

## На основе программы, составленной по лабораторной работе №3, выполнить статистический эксперимент по измерению фактической временной сложности алгоритма обработки данных, провести анализ.

## Описание эксперимента

Для проведения эксперимента программа из лабораторной работы №3 была модифицирована следующим образом:

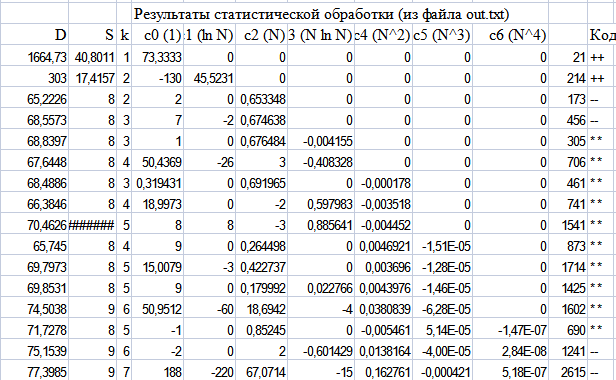
* Все операции с множествами и последовательностями выполнялись в цикле, изменяющим мощности множеств от 10 до 5000 элементов с шагом 10;
* Каждую итерацию производился замер времени выполнения операций с помощью библиотеки chrono;
* Для повышения достоверности эксперимента каждую итерацию определялась средняя мощность множества;

Теоретическая временная сложность определяется по «худшей» операции. Самое большее - O(n log n) - в операциях с последовательностями, где осуществляются произвольные вставки в дерево (с помощью функции Функция SUBST) сложностью O(log n) каждая.

**Результаты эксперимента**

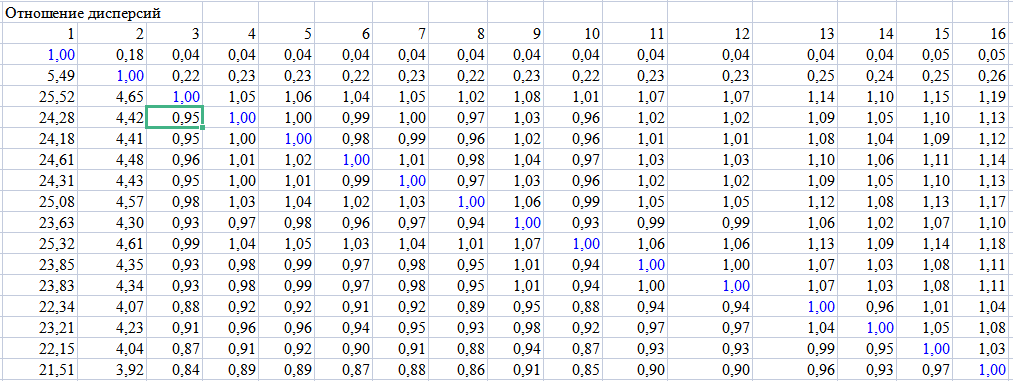
По результатам эксперимента был получен файл, содержащий 501 пару данных. Этот файл был обработан программой RG41 и результаты помещены в файл Microsoft Excel.

Результаты работы программы RG41 (501 точка)



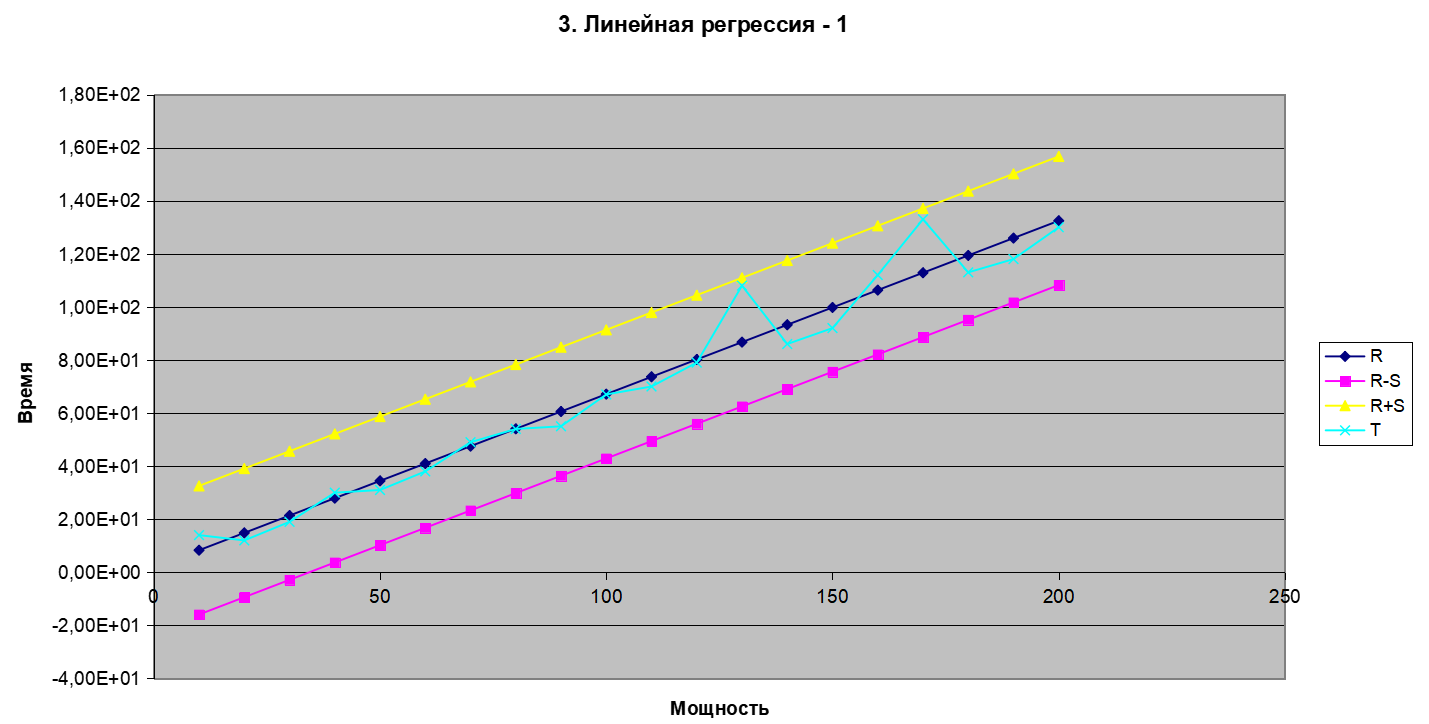
Результаты расчета отношений выборочных дисперсий

Отношение дисперсий



Проанализировав таблицы, мы можем сделать вывод, что соответствующая сложность “ Линейная регрессия-1 ” (O(n) - 1).

На графике мы видим, что практически все экспериментальные данные попадают в границы доверительного интервала ±3 СКО.



**Вывод**

В результате выполнения данной курсовой работы мы научились рассчитывать с помощью программы RG41 временную сложность и выяснили, что экспериментальная и теоретическая оценка временной сложности получились примерно схожи. Экспериментально подтвержденное значение временной сложности операций над множествами и последовательностями O(n) - 1.

**Текст программы#include <iostream>**#include <set>  
#include <vector>  
#include <chrono>  
#include <fstream>  
#include <algorithm>  
  
using MySet = std::set<int>;  
using MyIt = MySet::iterator;  
using MySeq = std::vector<MyIt>;  
  
int main() {  
 std::ofstream fout("output.txt");  
 const int num\_tests = 10; // Количество тестов  
 bool debug = false; // Отключаем отладочный вывод  
  
 for (int p = 10; p <= 200; p += 10) {  
 MySet myset;  
 MySeq myseq;  
  
 // Генерация множества  
 for (int i = 0; i < p; ++i) {  
 myset.insert(rand() % 1000); // Вставка случайных чисел  
 }  
  
 // Подготовка последовательности  
 for (auto it = myset.begin(); it != myset.end(); ++it) {  
 myseq.push\_back(it);  
 }  
  
 // Измерение времени выполнения операций  
 auto t1 = std::chrono::high\_resolution\_clock::now();  
  
 // Пример операций с множеством (объединение и пересечение)  
 MySet another\_set;  
 for (int i = 0; i < p / 2; ++i) {  
 another\_set.insert(rand() % 1000);  
 }  
  
 // Пример операции объединения  
 MySet result\_set;  
 std::set\_union(myset.begin(), myset.end(),  
 another\_set.begin(), another\_set.end(),  
 std::inserter(result\_set, result\_set.begin()));  
  
 // Пример операции пересечения  
 MySet intersection\_set;  
 std::set\_intersection(myset.begin(), myset.end(),  
 another\_set.begin(), another\_set.end(),  
 std::inserter(intersection\_set, intersection\_set.begin()));  
  
 auto t2 = std::chrono::high\_resolution\_clock::now();  
 auto duration = std::chrono::duration\_cast<std::chrono::microseconds>(t2 - t1).count();  
  
 fout << p << " " << duration << "\n"; // Запись результатов в файл  
  
 if (debug) {  
 std::cout << "Размер множества: " << p << ", Время: " << duration << " мкс" << std::endl;  
 }  
 }  
  
 fout.close();  
 return 0;  
}