|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ** |
| **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** |
| **«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** |
|  |
| Институт информационных технологий и управления в технических системах |
| (полное название института) |
|  |
| кафедра «Информационные системы» |
| (полное название кафедры) |

**ОТЧЁТ**

по лабораторной работе №2

на тему«ТЕОРЕТИЧЕСКАЯ ОЦЕНКА СРЕДНЕГО КОЛИЧЕСТВА

ОПЕРАЦИЙ ПЕРЕПРИСВАИВАНИЯ В АЛГОРИТМЕ ПОИСКА

МИНИМУМА»

Вариант 16

Выполнил

студент ИИТУТС

группы ИС/б-18-2-о

Радыгина Екатерина

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | |
|  | (должность, учёная степень преподавателя) | | | | | | | | |
|  |  | | | | | | | | |
|  | (ФИО преподавателя) | | | | | | | | |
|  | « |  | » |  |  |  | 20 | 20 | г. |
|  |  | | | | | | | | |
|  | (оценка) | | | | | | | | |
|  |  | | | | | | | | |
|  |  | | | | | | | | |

Севастополь 2020

**Лабораторная работа №2**

**«ТЕОРЕТИЧЕСКАЯ ОЦЕНКА СРЕДНЕГО КОЛИЧЕСТВА**

**ОПЕРАЦИЙ ПЕРЕПРИСВАИВАНИЯ В АЛГОРИТМЕ ПОИСКА**

**МИНИМУМА»**

**2.1 Цель работы**

Лабораторная работа посвящена экспериментальной проверке теоретической оценки трудоемкости алгоритма поиска минимума и включает ознакомление с принципами использования генератора случайных чисел для ёсоздания наборов исходных данных.

**2.2 Вариант задания**

Написать программу подсчета *n*-го гармонического числа. Подсчитать количество операций переприсваивания для программной реализации поиска минимума в массиве случайных чисел. Внести изменения в соответствующую программу. Длину массива и максимальное случайное число в последовательности взять в соответствии с вариантом.

Сравнить практически полученное значение с теоретическим *n*-м гармоническим числом. Исходные данные: наибольшее число в последовательности равно 200, а количество чисел в последовательности поочередно: 150, 550, 1000.

**2.3 Ход работы**

2.3.1 Функции для генерации псевдослучайных чисел

Функция rand генерирует случайные числа, возвращает псевдослучайное целое число в диапазоне. Это число генерируется алгоритмом, который возвращает последовательность псевдослучайных чисел. Этот алгоритм использует своего рода «семя» — число, для создания серий случайных чисел.

Функция random() возвращает случайное число в интервале от 0 до num-1.

Функция randomize() инициализирует генератор случайных чисел некоторой случайной величи­ной. Он использует функцию**time()**, так что необходимо включить в программу заголовочный файл **time.h**.

2.3.2 Для поиска минимального элемента был написан соответствующий алгоритм (рисунок 2.1).

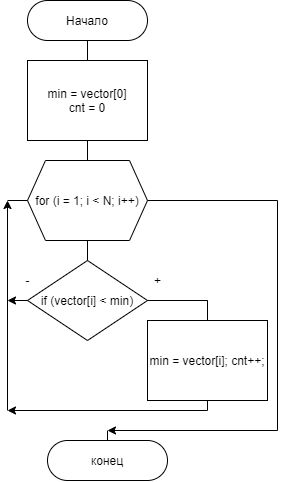
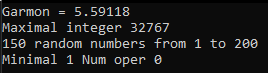
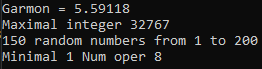


Рисунок 2.1 – Структурная схема алгоритма

2.3.3 Подсчёт количества переприсваиваний.

Количество элементов 150 (рисунок 2.2).





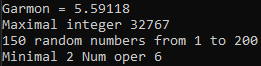
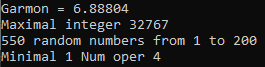
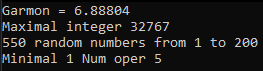


Рисунок 2.2 – Тесты при 150 элементах

Количество элементов 550 (рисунок 2.3).





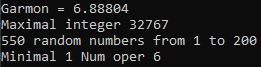
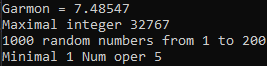
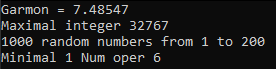


Рисунок 2.3 – Тесты при 550 элементах

Количество элементов 1000 (рисунок 2.4).





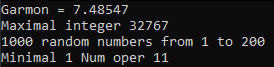


Рисунок 2.4 – Тесты при 1000 элементах

**Вывод**

В ходе данной работы была совершена экспериментальная проверка теоретической оценки трудоемкости алгоритма поиска минимума при помощи программы на языке С++. Были изучены генераторы случайных чисел и работа с ними в данном языке.

Также было обнаружено, что количество переприсваиваний при поиске минимального элемента в массиве редко превышает теоритическое n-ое гармоническое число.

Приложение А

#include <conio.h>

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <iomanip>

#include <time.h>

#include <iostream>

using namespace std;

const int N = 550;

int vector[1000];

void create\_array(int Nmax) {

int i;

FILE\* stream;

stream = fopen("Example\_TA2.TXT", "w+");

cout << "Maximal integer "; cout << RAND\_MAX;

printf("\n%d%s\n", Nmax, " random numbers from 1 to 200");

for (i = 0; i < Nmax; i++) {

vector[i] = rand() % 200 + 1;

fprintf(stream, "%d\n", vector[i]);

}

fclose(stream);

}

double harmonic(int N) {

double Sum = 0;

for (int i = 1; i <= N; i++) { Sum += 1. / i; }

return Sum;

}

int main() {

int i,min,cnt;

double result;

srand(time(0));

result = harmonic(N);

cout << "Garmon = " << result << endl;

create\_array(N);

min = vector[0];

cnt = 0;

for (i = 1; i < N; i++) {

if (vector[i] < min) { min = vector[i]; cnt++; }

}

printf("%s%d%s%d\n", "Minimal ", min, " Num oper ", cnt);

}