

ЛАБОРАТОРНАЯ РАБОТА № 2.  
“ТЕОРЕТИЧЕСКАЯ ОЦЕНКА СРЕДНЕГО КОЛИЧЕСТВА  
ОПЕРАЦИЙ ПЕРЕПРИСВАИВАНИЯ В АЛГОРИТМЕ ПОИСКА  
МИНИМУМА”

## 1. ЦЕЛЬ РАБОТЫ

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

## 2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

2.1. Теоретическая оценка среднего количества операций переприсваивания в алгоритме поиска минимума

Алгоритм поиска минимума последовательно перебирает элементы массива, сравнивая текущий элемент массива с текущим значением минимума. На очередном шаге, когда просматривается  $k$ -ый элемент массива, переприсваивание минимума произойдет, если в подмассиве из первых  $k$  элементов минимальным элементом является последний. В случае равномерного распределения исходных данных вероятность того, что максимальный из  $k$  элементов расположен в определенной (последней) позиции равна  $1/k$ .

Тогда в массиве из  $N$  элементов среднее количество операций переприсваивания максимума определяется как:

$$\sum_{i=1}^N 1/i = H_n \approx \ln(N) + \gamma, \text{ где } \gamma = 0,57$$

Величина  $H_n$  называется  $n$ -ым гармоническим числом. Таким образом среднее значение (математическое ожидание) среднего количества операций присваивания в алгоритме поиска минимума в массиве из  $N$  элементов определяется величиной  $H_n$  (для бесконечно большого количества испытаний).

## 2.2. Понятие о генераторе псевдослучайных чисел

Для тестирования программ часто необходимо использовать наборы случайных чисел. Такие числа могут быть использованы и для реализации тех или иных алгоритмов. Для создания таких чисел используется генератор случайных чисел. В общем случае алгоритм называется рандомизированным (randomized), если его поведение определяется не только набором входных

величин, но и значениями, которые выдает генератор случайных чисел.

На практике большинство сред программирования предоставляют в распоряжение программиста генератор псевдослучайных чисел, т.е. детерминированный алгоритм, который возвращающий числа, которые ведут себя при статистическом анализе как случайные.

### 3. ХОД РАБОТЫ

1. Ознакомиться с теоретическим разделом настоящих методических указаний и повторить соответствующий лекционный материал.

2. Изучить функции языка C++, использующиеся для генерации псевдослучайных чисел, привести их описание в отчете (rand, random, randomize и др.).

3. Составить структурную схему и написать программу поиска минимума в массиве сгенерированных псевдослучайных чисел.

4. Написать программу подсчета  $n$ -го гармонического числа.

5. 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.

6. Сравнить практически полученное значение с теоретическим  $n$ -м гармоническим числом. Примеры основных функций приведены в приложении.

7. Сделать выводы по работе, оформить отчет, подготовить ответы на контрольные вопросы.

#### 4. ВАРИАНТЫ ЗАДАНИЙ

Вариант	Наибольшее случайное число в последовательности	Количество элементов в массиве случайных чисел
1	100	100, 1000, 5000
2	250	150, 550, 1000
3	150	100, 500, 1000
4	950	250, 300, 400
5	200	200, 300, 500
6	800	100, 1000, 2000
7	250	300, 400, 1400
8	850	300, 450, 1000
9	200	150, 200, 1000
10	350	100, 500, 1500
11	100	300, 900, 3000
12	400	100, 200, 500
13	500	100, 1000, 1500
14	950	250, 500, 750
15	150	100, 1000, 5000
16	200	150, 550, 1000
17	750	100, 500, 1000
18	550	250, 300, 400
19	150	200, 300, 500
20	200	100, 1000, 2000

#### 5. ПРИМЕРЫ ОСНОВНЫХ ФУНКЦИЙ, ИСПОЛЬЗУЕМЫХ В ПРОГРАММЕ

```
#include <conio.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <iomanip.h>
#include <time.h>
```

```
int vector[10];
// Создание массива из 10 псевдослучайных целых чисел величиной
от 0 до 100
// массив записывается в файл Example_TA2.TXT, на экран выводим
максимальное // целое
```

```
void create_array(int Nmax) {
    int i;
```

```

FILE *stream;
//Nmax = 10; соответствует размерности массива
stream = fopen("Example_TA2.TXT", "w+");
randomize();
cout<<"Maximal integer ";   cout<<RAND_MAX;
printf("\n%d%s\n", Nmax, " random numbers from 0 to 100");
for(i=0; i<Nmax; i++){
    vector[i]= rand() % 100;
    printf("%d\n", vector[i]);
    fprintf(stream,"%d\n", vector[i]);
}
fclose(stream);
}

```

```

void main(){

    int    i,
           N,
           min, // значение минимума
           cnt; //счетчик операций переприсваивания

    double result;

    cout<<"Input amount of numbers"; cin>>N;
    result = harmonic(N);
           // harmonic(N) – функция подсчета n-го гармонического
числа
    cout<<result;
    create_array(N);
           //генерация массива псевдослучайных чисел
    min = vector[0];
    cnt = 1;
    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);
}

```

## 6. СОДЕРЖАНИЕ ОТЧЁТА

1. Цель работы.
2. Вариант задания.
3. Тексты программы, реализующих расчеты по соответствующему варианту, описания функций, используемых для генерации наборов исходных данных, структурные схемы основных функций, используемых в программе.
4. Результаты работы программы.
5. Развернутый вывод по работе.

## 7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Понятие о трудоемкости алгоритма.
2. Понятие о сложности алгоритма.
3. Определение детерминированного и рандомизированного алгоритмов.
4. Определение генератора псевдослучайных чисел.
5. Описать основные функции и константы, используемые для создания последовательности псевдослучайных чисел в языке C++.