МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И. С. ТУРГЕНЕВА»

                                                                                                                                                                   Кафедра информационных систем и цифровых технологий

**ОТЧЁТ**

по лабораторной работе № 5

на тему: «Автоматизированное тестирование программного обеспечения»

по дисциплине «Качество и тестирование программного обеспечения»

Выполнили: Банных М.А., Мельников А.Е.

Институт приборостроения, автоматизации и информационных

технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 21ПГ

Проверил: Олькина Е.В.

Отметка о зачёте:

Дата: «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.

Орёл, 2025

**Листинг программы:**

#include <algorithm>

using namespace std;

int compareInts(const void\* a, const void\* b)

{

int arg1 = \*(const int\*)a;

int arg2 = \*(const int\*)b;

if (arg1 < arg2) return -1;

if (arg1 > arg2) return 1;

return 0;

}

int countTransmitters(int\* arr, int n, int k) {

if (n <= 0 || k <= 0) {

throw exception("Неправильные исходные данные");

}

qsort(arr, n, sizeof(int), compareInts);

if (arr[0] <= 0) {

throw exception("Неправильные исходные данные");

}

int count = 0;

int i = 0;

while (i < n) {

count++;

int coverage = arr[i] + k;

while (i < n && arr[i] <= coverage)

i++;

coverage = arr[i - 1] + k;

while (i < n && arr[i] <= coverage)

i++;

}

return count;

}

**Потоковый граф**

Вычисление цикломатической сложности:

1. 24 дуг – 19 узлов + 2 = 7

2. 6 предикатных узлов + 1 = 7

3. 7 регионов

Пути:

1-3

1-2-3

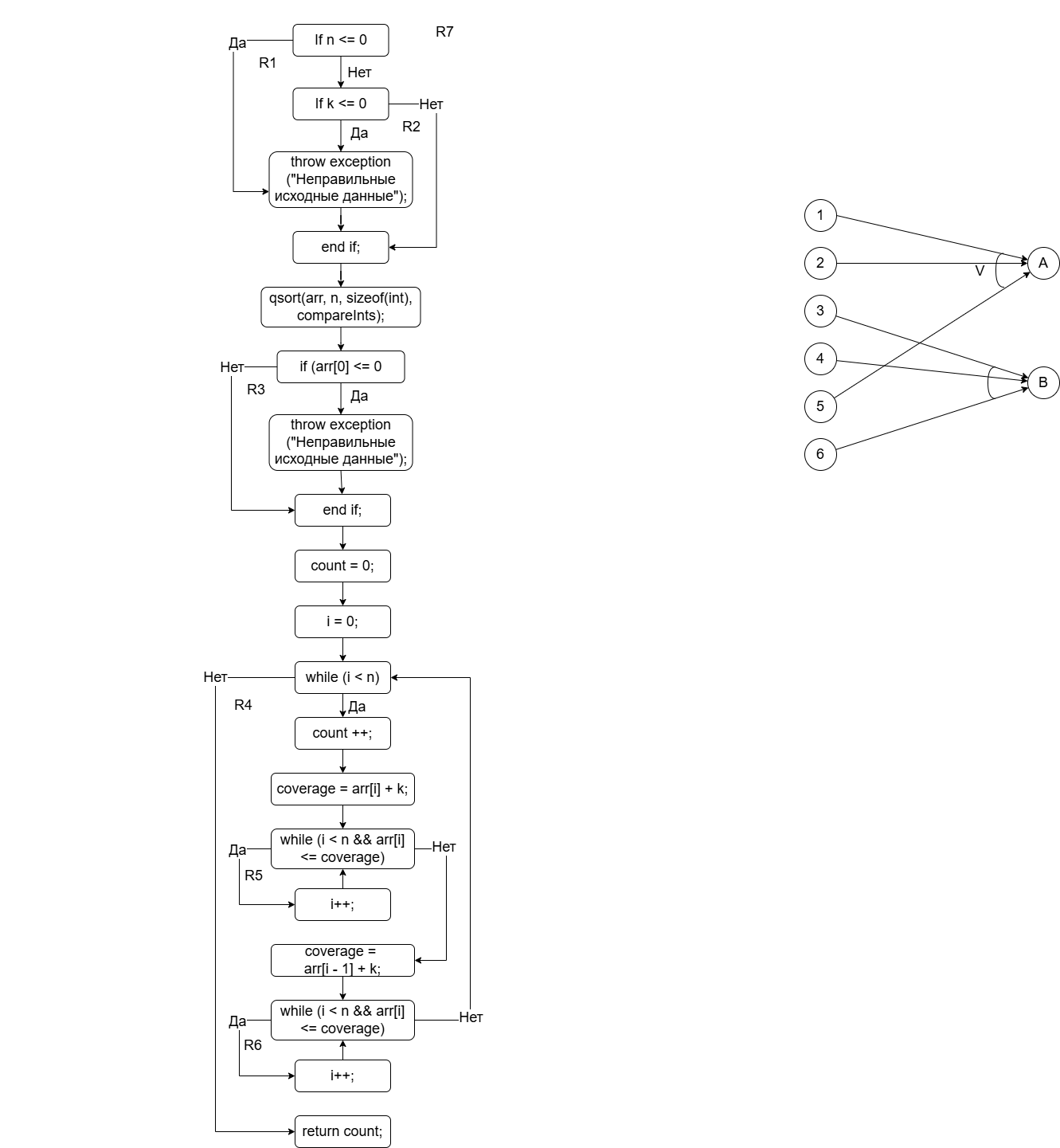
1-2-4-5-6-7

1-2-4-5-6-8-9-10-11-12-13-14-15-16-17-18-19

1-2-4-5-6-8-9-10-11-12-13-14-15-16-17-19

1-2-4-5-6-8-9-10-11-12-13-14-16-17-19 (невозможен по выражению 10 и условию 2)

1-2-4-5-6-8-9-10-11-19 (невозможен по условию 1)

****

**Граф причин-следствий**

Причины:

1. n меньше или равно 0

2. k меньше или равно 0

3. n больше или равно 1

4. k больше или равно 1

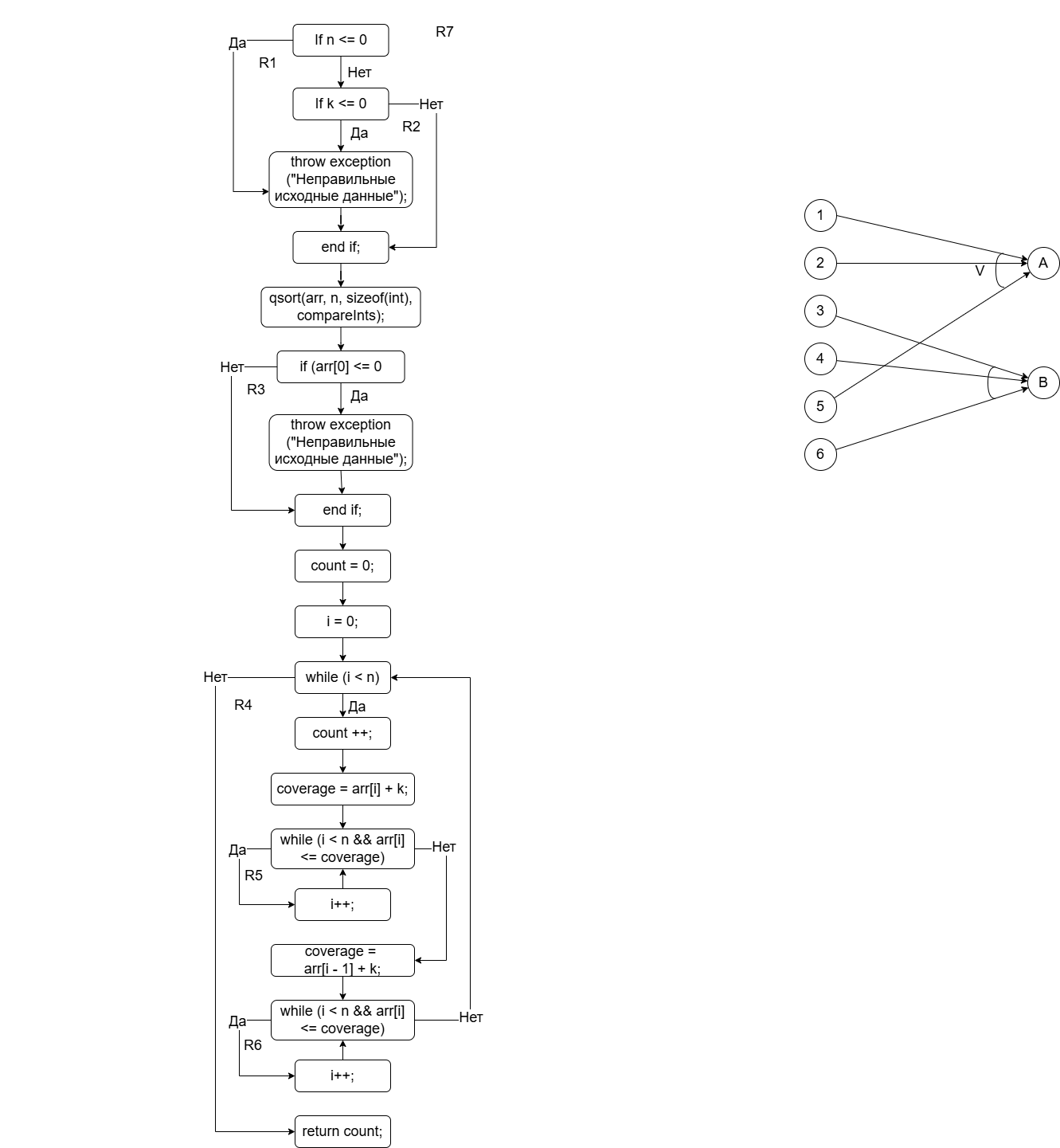
5. Существует дом с координатой меньшей или равной 0

6. Все дома с положительными координатами

Следствия:

A. Неправильные исходные данные

B. Вывод минимального числа передатчиков



**Тесты**

#include "pch.h"

#include "CppUnitTest.h"

#include "../QualityAndTesting 5 Lab/QualityAndTesting 5 Lab.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest

{

TEST\_CLASS(UnitTest)

{

public:

TEST\_METHOD(TestN0)

{

int k = 6;

int n = 0;

int\* arr = nullptr;

string expected = "Неправильные исходные данные";

string result;

try {

countTransmitters(arr, n, k);

}

catch (exception e) {

result = e.what();

}

Assert::AreEqual(expected, result);

}

TEST\_METHOD(TestK0)

{

int k = 0;

int n = 5;

int arr[5] = { 3, 1, 4, 6, 2 };

string expected = "Неправильные исходные данные";

string result;

try {

countTransmitters(arr, n, k);

}

catch (exception e) {

result = e.what();

}

Assert::AreEqual(expected, result);

}

TEST\_METHOD(TestExistXNonPositive)

{

int k = 6;

int n = 5;

int arr[5] = { 3, -1, 4, 0, 2 };

string expected = "Неправильные исходные данные";

string result;

try {

countTransmitters(arr, n, k);

}

catch (exception e) {

result = e.what();

}

Assert::AreEqual(expected, result);

}

TEST\_METHOD(TestTwoHousesInRow)

{

int k = 4;

int n = 3;

int arr[3] = { 19, 23, 24 };

int expected = 1;

int result = countTransmitters(arr, n, k);

Assert::AreEqual(expected, result);

}

TEST\_METHOD(TestNoTwoHousesInRow)

{

int k = 7;

int n = 2;

int arr[3] = { 1, 9 };

int expected = 2;

int result = countTransmitters(arr, n, k);

Assert::AreEqual(expected, result);

}

};

}