

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Качество и метрология программного обеспечения»

Тема: «Анализ структурной сложности графовых моделей программ»

Студент гр. 8304

Сергеев А.Д.

Преподаватель

Кирияничиков В.А.

Санкт-Петербург

2022

Формулировка задания

Выполнить оценивание структурной сложности двух программ с помощью критериев:

- Минимального покрытия дуг графа;
- Выбора маршрутов на основе цикломатического числа графа.

Варианты программ:

- Программа с заданной преподавателем структурой управляющего графа;
- Программа из 1-ой лабораторной работы (управляющий граф составить самостоятельно).

Оцениваемые характеристики структурной сложности:

- Число учитываемых маршрутов проверки программы для заданного критерия;
- Цикломатическое число;
- Суммарное число ветвлений по всем маршрутам.

Ход работы

1. Анализ заданной программы (вариант 18)

1.1. Граф программы 1

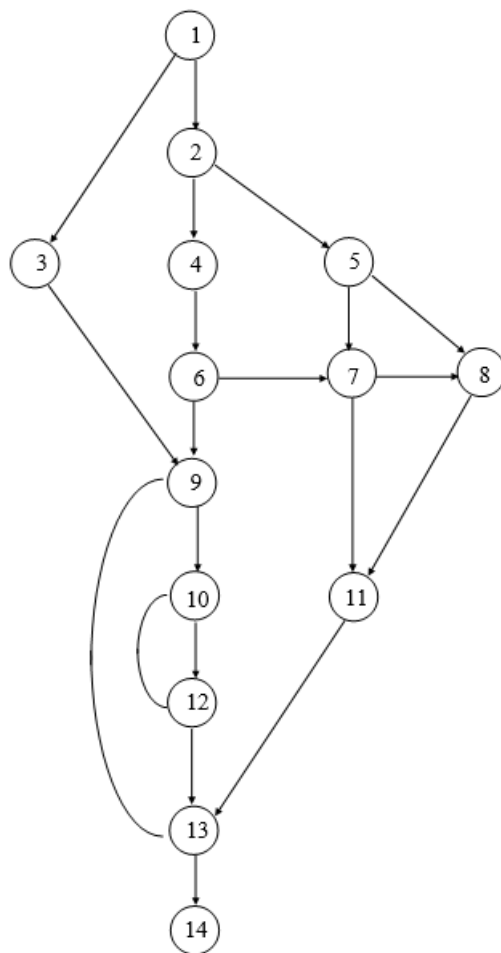


Рисунок 1 – Заданный управляющий граф

1.2. Ручные расчеты

Критерий 1:

М1: 1-2-5 -8-11- 13 -14	4 ветвлений
М2: 1-2-5-7 -8-11- 13 -14	5 ветвления
М3: 1-2-4-6 -9-10- 12-13 -14	5 ветвления
М4: 1-3-9-10-12-13 -14	3 ветвления
М5: 1-2-4-6-7-11-13-9-10-12-10-12-13 -14	8 ветвления

Количество маршрутов $M = 5$

Сложность: $S_2 = \sum_{i=1}^M \xi_i = 4 + 5 + 5 + 3 + 8 = 25$

Критерий 2:

Добавим ребро 14-1 для того чтобы граф стал связным.

$$Z = Y - N + 2 * P = 20 - 14 + 2 * 1 = 8$$

Количество проверяемых маршрутов равно цикломатическому числу графа.

M1: 1-2-5-8-11-13-14	4 ветвлений
M2: 1-2-5-7-8-11-13-14	5 ветвления
M3: 1-2-4-6-9-10-12-13-14	5 ветвления
M4: 1-3-9-10-12-13-14	3 ветвления
M5: 1-2-4-6-7-11-13-14	5 ветвления
M6: 1-2-4-6-7-8-11-13-14	5 ветвления
M7: 10- 12 -10	1 ветвления
M8: 9-10- 12-13 -9	2 ветвления

$$\text{Сложность: } S_2 = \sum_{i=1}^M \xi_i = 4 + 5 + 5 + 3 + 5 + 5 + 1 + 2 = 30$$

1.3. Автоматические расчёты

Описание структуры графа:

Nodes{ 1,2,3,4,5,6,7,8,9,10,11,12,13,14}

Top{ 1}

Last{ 14}

Arcs{

arc(1,2);

arc(1,3);

arc(2,4);

arc(2,5);

arc(3,9);

arc(4,6);

arc(5,7);

arc(5,8);

arc(6,7);

arc(6,9);

arc(7,8);

arc(7,11);

arc(8,11);

arc(9,10);

arc(10,12);

arc(11,13);

arc(12,13);

arc(12,10);

arc(13,14);

arc(13,9);

}

Результат анализа структурной сложности:

```
Min ways....
----- Path #1 -----
-> 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 11 -> 13 -> 9 -> 10 -> 12 -> 10 -> 12 -> 13 ->
14
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 3 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 5 -> 7 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 4 -> 6 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #5 -----
-> 1 -> 2 -> 5 -> 8 -> 11 -> 13 -> 14
-----Press a key to continue -----

Complexity = 25
Press a key...
```

Рисунок 2 - Результат выполнения ways.exe для 1-го критерия

```
Z ways....
----- Path #1 -----
-> 10 -> 12 -> 10
-----Press a key to continue -----
----- Path #2 -----
-> 9 -> 10 -> 12 -> 13 -> 9
-----Press a key to continue -----
----- Path #1 -----
-> 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 4 -> 6 -> 7 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 4 -> 6 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 5 -> 7 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #5 -----
-> 1 -> 2 -> 5 -> 8 -> 11 -> 13 -> 14
-----Press a key to continue -----
----- Path #6 -----
-> 1 -> 3 -> 9 -> 10 -> 12 -> 13 -> 14
-----Press a key to continue -----

Complexity = 30
Press a key...
```

Рисунок 3 - Результат выполнения ways.exe для 2-го критерия

2. Анализ программы из 1-ой лабораторной работы

2.1. Граф программы 2

Код программы представлен в приложении А.

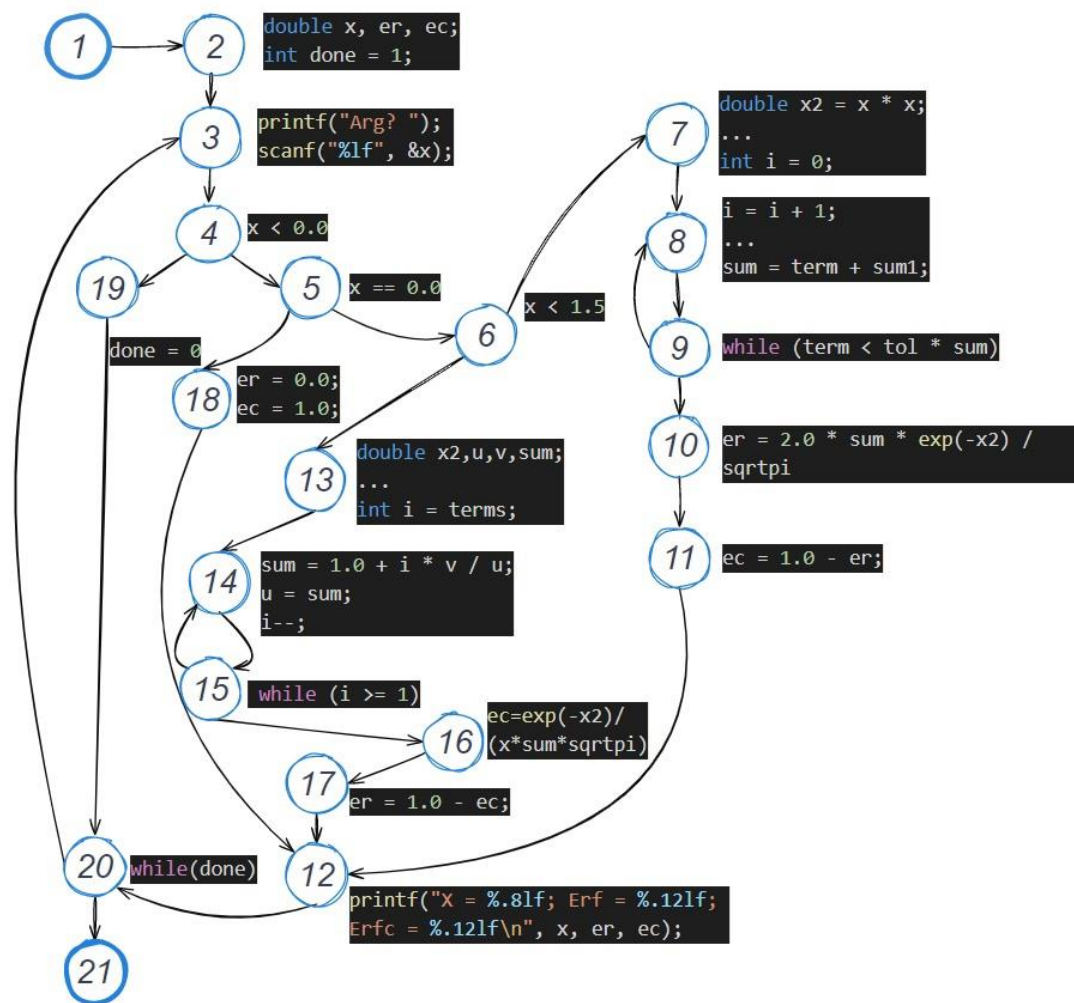


Рисунок 6 – управляющий граф программы 2

2.2. Ручные расчеты

Критерий 1:

М1: 1-2-3-**4-5-6**-7-8-**9-8-9**-10-11-12-**20-3-4-5-6**-13-14-**15-14-15**-16-17-12-**20-3-4-5**-18-12-**20-3-4**-19-**20-21**

Итого 17 ветвлений.

Количество маршрутов $M = 1$

Сложность: $S=17$.

Критерий 2:

$$Z = Y - N + 2 * P = 26 - 21 + 2 * 1 = 7$$

Количество проверяемых маршрутов равно цикломатическому числу графа.

8-**9**-8

14-**15**-14

3-**4**-19-**20**-3

3-**4-5**-18-12-**20**-3

3-**4-5-6**-13-14-**15**-16-17-12-**20**-3

3-**4-5-6**-7-8-**9**-10-11-12-**20**-3

1-2-3-**4**-19-**20**-21

Итого 19 ветвлений.

2.3. Автоматические расчёты

Nodes{ 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21 }

Top{ 1 }

Last{ 21 }

Arcs{

arc(1,2);

arc(2,3);

arc(3,4);

arc(4,5);

arc(4,19);

arc(5,6);

arc(5,18);

arc(6,7);

arc(6,13);

arc(7,8);

arc(8,9);

arc(9,8);

arc(9,10);

arc(10,11);

arc(11,12);

arc(12,20);

arc(13,14);

arc(14,15);

arc(15,14);

arc(15,16);

arc(16,17);

arc(17,12);

arc(18,12);

arc(19,20);

arc(20,21);

arc(20,3);

}

Результат анализа структурной сложности:

```
Min ways...
----- Path #1 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 8 -> 9 -> 10 -> 11 -> 12 -> 20
-> 3 -> 4 -> 19 -> 20 -> 3 -> 4 -> 5 -> 18 -> 12 -> 20 -> 3 -> 4 -> 5 -> 6 -> 13
-> 14 -> 15 -> 14 -> 15 -> 16 -> 17 -> 12 -> 20 -> 21
-----Press a key to continue -----

Complexity = 17
Press a key...
```

Рисунок 7 - Результат выполнения ways.exe для 1-го критерия

```

----- Path #1 -----
-> 8 -> 9 -> 8
-----Press a key to continue -----
----- Path #2 -----
-> 14 -> 15 -> 14
-----Press a key to continue -----
----- Path #3 -----
-> 3 -> 4 -> 19 -> 20 -> 3
-----Press a key to continue -----
----- Path #1 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10 -> 11 -> 12 -> 20 -> 21
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 13 -> 14 -> 15 -> 16 -> 17 -> 12 -> 20 -> 21
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 18 -> 12 -> 20 -> 21
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 3 -> 4 -> 19 -> 20 -> 21
-----Press a key to continue -----

Complexity = 19
Press a key...

```

Рисунок 8 - Результат выполнения ways.exe для 2-го критерия

Вывод

В данной лабораторной работе была выполнена оценка структурной сложности двух программ с по двум критериям: минимальному покрытию дуг их графа и выбору маршрутов на основе цикломатического числа их графа. Расчеты были проведены как ручным, так и программным способом.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ИЗ ЛАБОРАТОРНОЙ №1

```
#include <math.h>
#include <stdio.h>

const double sqrtpi = 1.7724538;
const double tol = 1.0E-4;
const int terms = 12;

// infinite series expansion of the Gaussian error function
double erf (double x) {
    double x2 = x * x;
    double sum = x;
    double term = x;
    int i = 0;
    do {
        i = i + 1;
        double sum1 = sum;
        term = 2.0 * term * x2 / (1.0 + 2.0 * i);
        sum = term + sum1;
    } while (term < tol * sum);
    return 2.0 * sum * exp(-x2) / sqrtpi;
}

// complement of error function
double erfc (double x) {
    double x2,u,v,sum;
    x2 = x * x;
    v = 1.0 / (2.0 * x2);
    u = 1.0 + v * (terms + 1.0);
    int i = terms;
    do {
        sum = 1.0 + i * v / u;
        u = sum;
        i--;
    } while (i >= 1);
    return exp(-x2) / (x * sum * sqrtpi);
}

// evaluation of the gaussian error function
int main () {
    double x, er, ec;
    int done = 1;
    do {
        printf("Arg? ");
        scanf("%lf", &x);
        if (x < 0.0) done = 0;
        else {
            if (x == 0.0) {
                er = 0.0;
                ec = 1.0;
            } else {
                if (x < 1.5) {
                    er = erf(x);
                    ec = 1.0 - er;
                } else {
                    ec = erfc(x);
                    er = 1.0 - ec;
                }
            }
        }
    } while (done);
}
```

```
        }
    }
    printf("X = %.8lf; Erf = %.12lf; Erfc = %.12lf\n", x, er, ec);
}
} while (done);
}
```