

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

**ОТЧЕТ
по лабораторной работе №2
по дисциплине «Качество и метрология программного обеспечения»
Тема: Анализ структурной сложности графовых моделей программ**

Студентка гр. 7304 _____

Юреть Е.А.

Преподаватель _____

Ефремов М.А.

Санкт-Петербург

2021

Цель работы

Изучение применения метрик структурной сложности программ — критерия минимального покрытия и анализа базовых маршрутов.

Постановка задачи

Выполнить оценивание структурной сложности двух программ с помощью критериев:

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

Варианты программ:

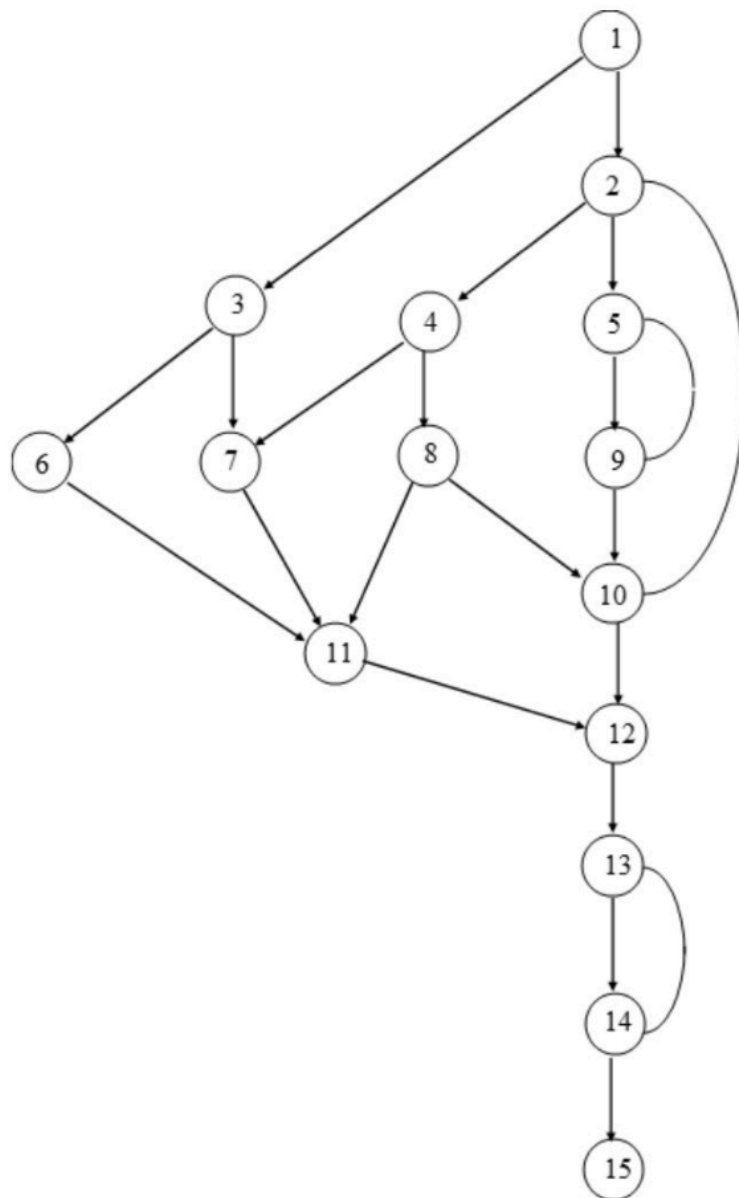
- Программа с заданной структурой управляющего графа, выбираемой из файла `zadan_struct.doc` в соответствии с номером в списке группы;
- Программа из 1-ой лабораторной работы (управляющий граф составить самостоятельно).

Оцениваемые характеристики структурной сложности:

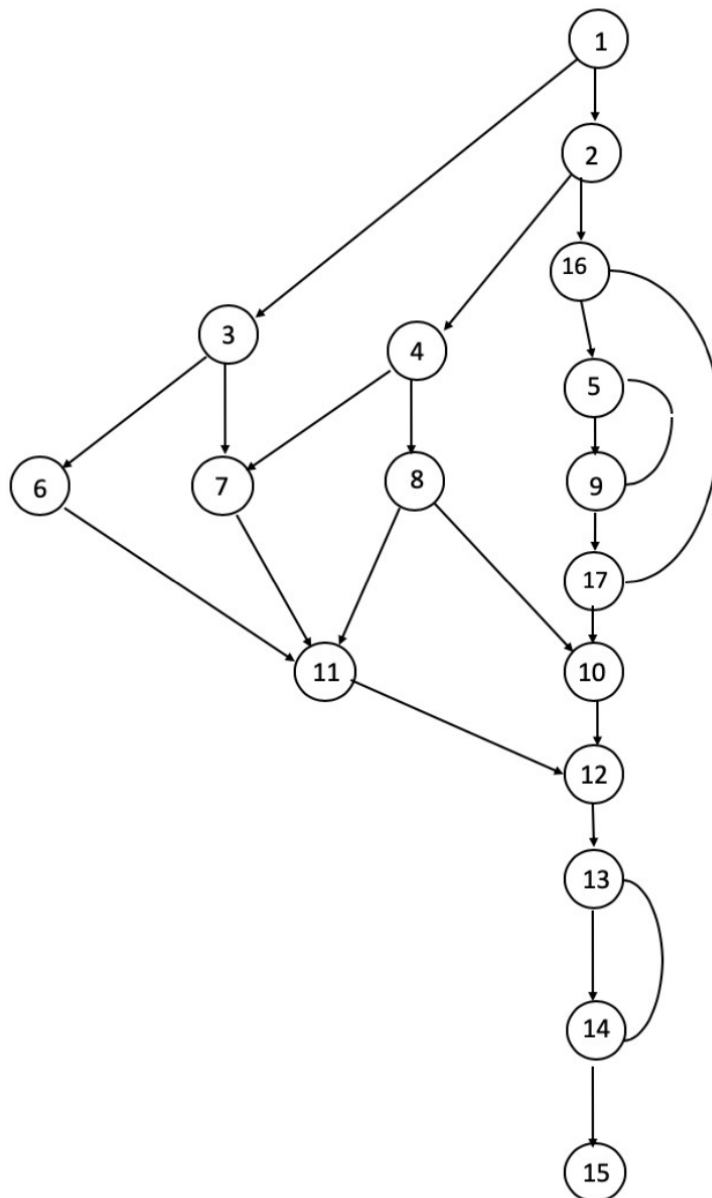
- Число учитываемых маршрутов проверки программы для заданного критерия;
- Цикломатическое число;
- Суммарное число ветвлений по всем маршрутам — оценка структурной сложности;

Ход работы

Исходный граф:



Модифицированный граф:



1.1. Оценивание структурной сложности с помощью критерия:

минимальное покрытие вершин и дуг графа управления:

$Y = 24$ – общее число дуг графа;

$N = 17$ – общее число вершин графа;

$\Omega = 1$ – число связных компонент

$n_b = 8$ – число вершин, в которых происходит ветвление.

$$Z = Y - N + 2 * \Omega = 24 - 17 + 2 * 1 = 8$$

$$Z = n_b + 1 = 8 + 1 = 9$$

M1: **1-3-6-11-12-13-14-15**

M2: **1-3-7-11-12-13-14-15**

M3: **1-2-4-7-11-12-13-14-15**

M4: **1-2-4-8-11-12-13-14-15**

M5: 1-2-4-8-10-12-13-14-15

M6: 1-2-16-5-9-5-9-17-16-5-9-17-10-12-13-14-13-14-15

S = 29

- 1.2. Оценивание структурной сложности с помощью критерия: выбор маршрутов на основе цикломатического числа графа:

$$Z = n_B + 1 = 8 + 1 = 9$$

M1: 1-3-6-11-12-13-14-15

M2: 1-3-7-11-12-13-14-15

M3: 1-2-4-7-11-12-13-14-15

M4: 1-2-4-8-11-12-13-14-15

M5: 1-2-4-8-10-12-13-14-15

M6: 1-2-5-9-17-10-12-13-14-15

M7: 5-9

M8: 13-14

M9: 2-5-9-17

S = 29

- 1.3. Программный расчет:

```
Min ways....
----- Path #1 -----
-> 1 -> 2 -> 16 -> 5 -> 9 -> 5 -> 9 -> 17 -> 16 -> 5 -> 9 -> 17 -> 10 -> 12 ->
13 -> 14 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #2 -----
-> 1 -> 3 -> 6 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 4 -> 7 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 4 -> 8 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #5 -----
-> 1 -> 2 -> 4 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #6 -----
-> 1 -> 3 -> 7 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
Complexity = 29
Press a key...
```

```
Z ways....
----- Path #1 -----
-> 5 -> 9 -> 5
----- Press a key to continue -----
----- Path #2 -----
-> 16 -> 5 -> 9 -> 17 -> 16
----- Press a key to continue -----
----- Path #3 -----
-> 13 -> 14 -> 13
```

```

----- Path #1 -----
-> 1 -> 2 -> 16 -> 5 -> 9 -> 17 -> 10 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 4 -> 7 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 4 -> 8 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 4 -> 8 -> 10 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #5 -----
-> 1 -> 3 -> 6 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
----- Path #6 -----
-> 1 -> 3 -> 7 -> 11 -> 12 -> 13 -> 14 -> 15
----- Press a key to continue -----
Complexity = 29
Press a key...

```

```

program erfd4;
uses Crt;
{ evaluation of the gaussian error function }

var      x,er,ec          : real;
        done              : boolean;

function erf(x: real): real;
{ infinite series expansion of the Gaussian error function }

const    sqrtpi           = 1.7724538;
        t2                = 0.66666667;
        t3                = 0.66666667;
        t4                = 0.07619048;
        t5                = 0.01693122;
        t6                = 3.078403E-3;
        t7                = 4.736005E-4;
        t8                = 6.314673E-5;
        t9                = 7.429027E-6;
        t10               = 7.820028E-7;
        t11               = 7.447646E-8;
        t12               = 6.476214E-9;

var      x2,sum            : real;
        i                  : integer;

begin
    x2:=x*x;

    sum:=t5+x2*(t6+x2*(t7+x2*(t8+x2*(t9+x2*(t10+x2*(t11+x2*t12))))));

```

```

    erf:=2.0*exp(-x2)/sqrtpi*(x*(1+x2*(t2+x2*(t3+x2*(t4+x2*sum))))
end;      { function erf }

function erfc(x: real): real;
{ complement of error function }
const    sqrtpi          = 1.7724538;

var      x2,v,sum      : real;

begin
    x2:=x*x;
    v:=1.0/(2.0*x2);
    sum:=v/(1+8*v/(1+9*v/(1+10*v/(1+11*v/(1+12*v)))));
    sum:=v/(1+3*v/(1+4*v/(1+5*v/(1+6*v/(1+7*sum)))));
    erfc:=1.0/(exp(x2)*x*sqrtpi*(1+v/(1+2*sum)))
end;      { function ercf }

begin      { main }
    done:=false;
    writeln;
    repeat
        write('Arg? ');
        readln(x);
        if x<0.0 then done:=true
        else
            begin
                if x=0.0 then
                    begin
                        er:=0.0;
                        ec:=1.0
                    end
                else
                    begin
                        if x<1.5 then
                            begin
                                er:=erf(x);
                                ec:=1.0-er
                            end
                        else
                            begin
                                ec:=erfc(x);
                                er:=1.0-ec
                            end
                        end
                    end
                end
            end
        end
    end

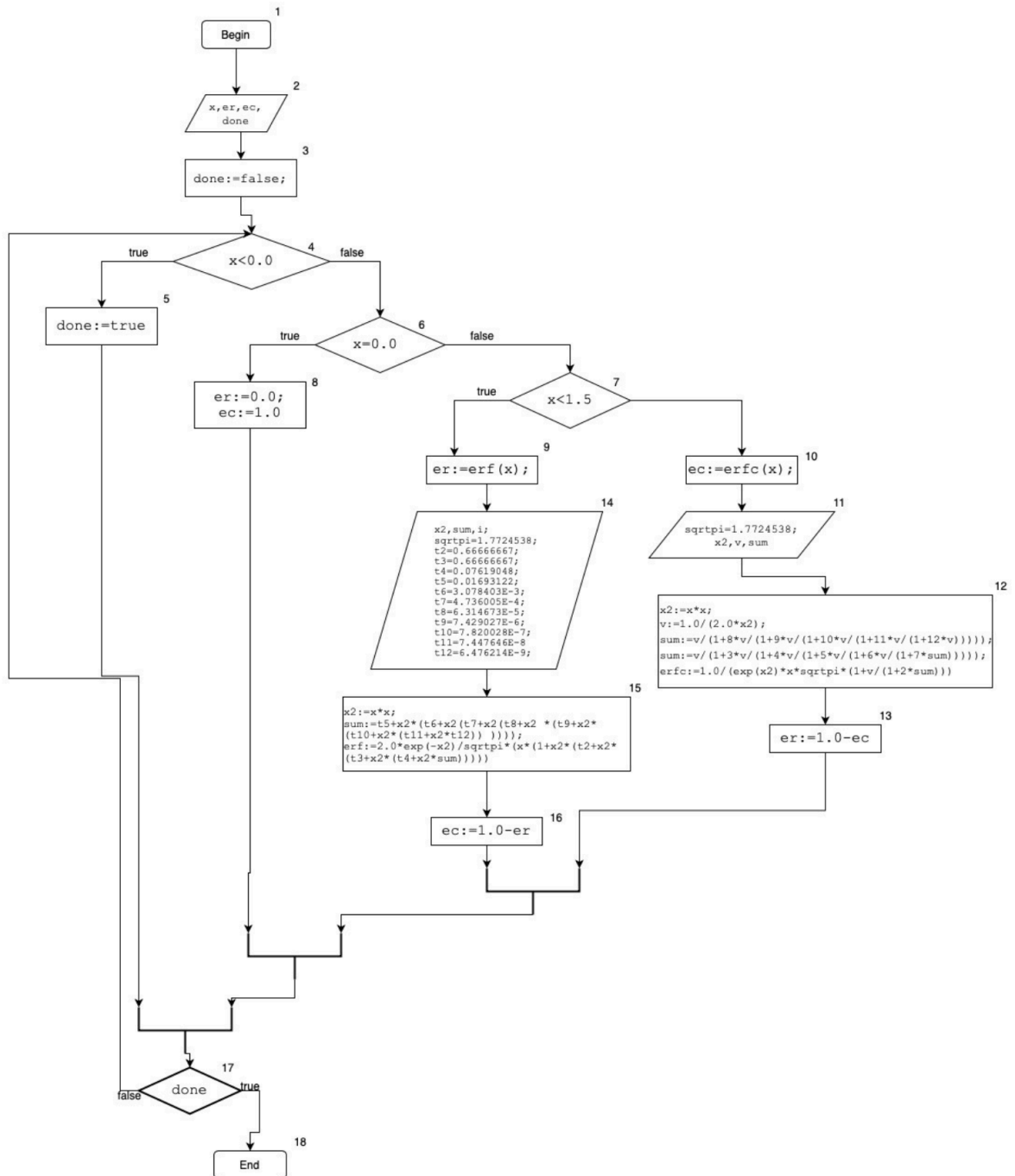
```

```

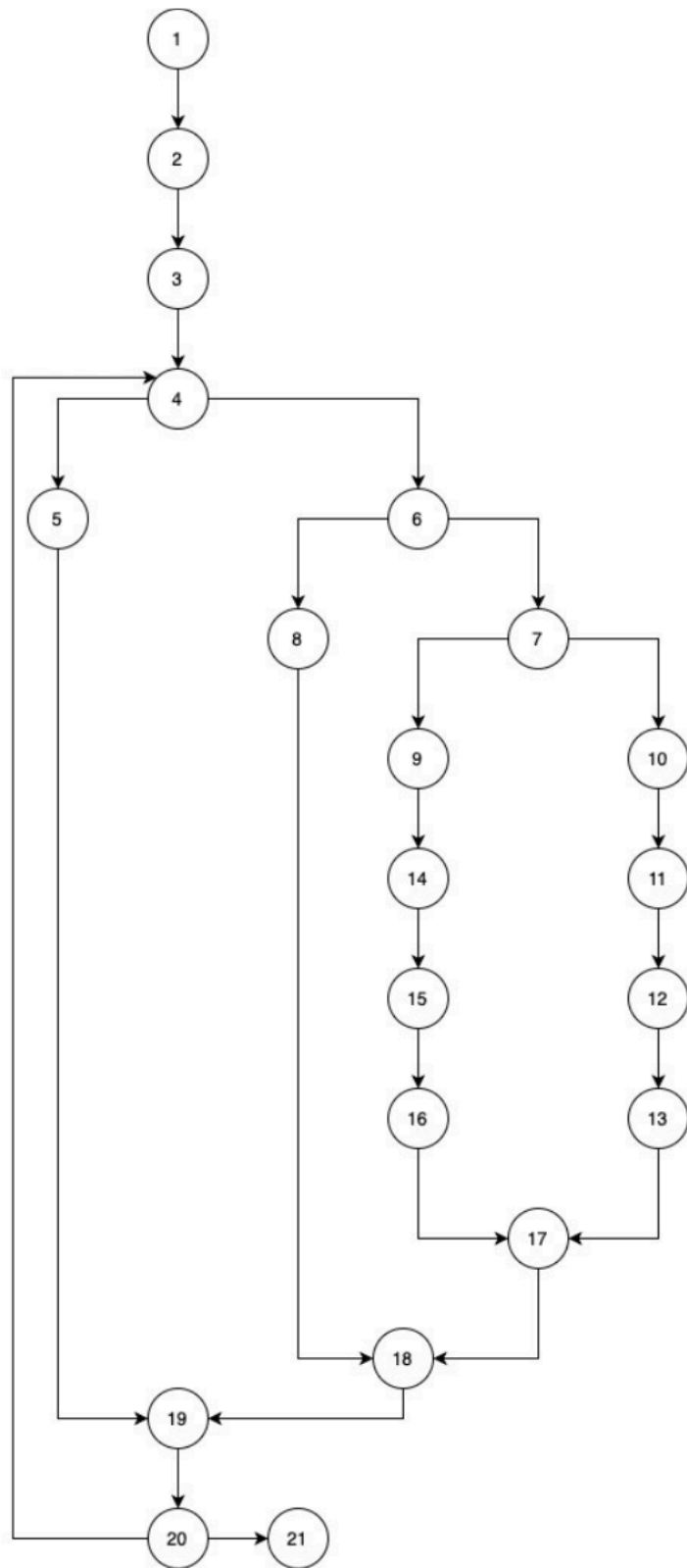
        end    { if }
    end;
    writeln('X= ',x:8:4,', Erf= ',er:12,', Erfc= ',ec:12)
        end    { if }
    until done
end.

```

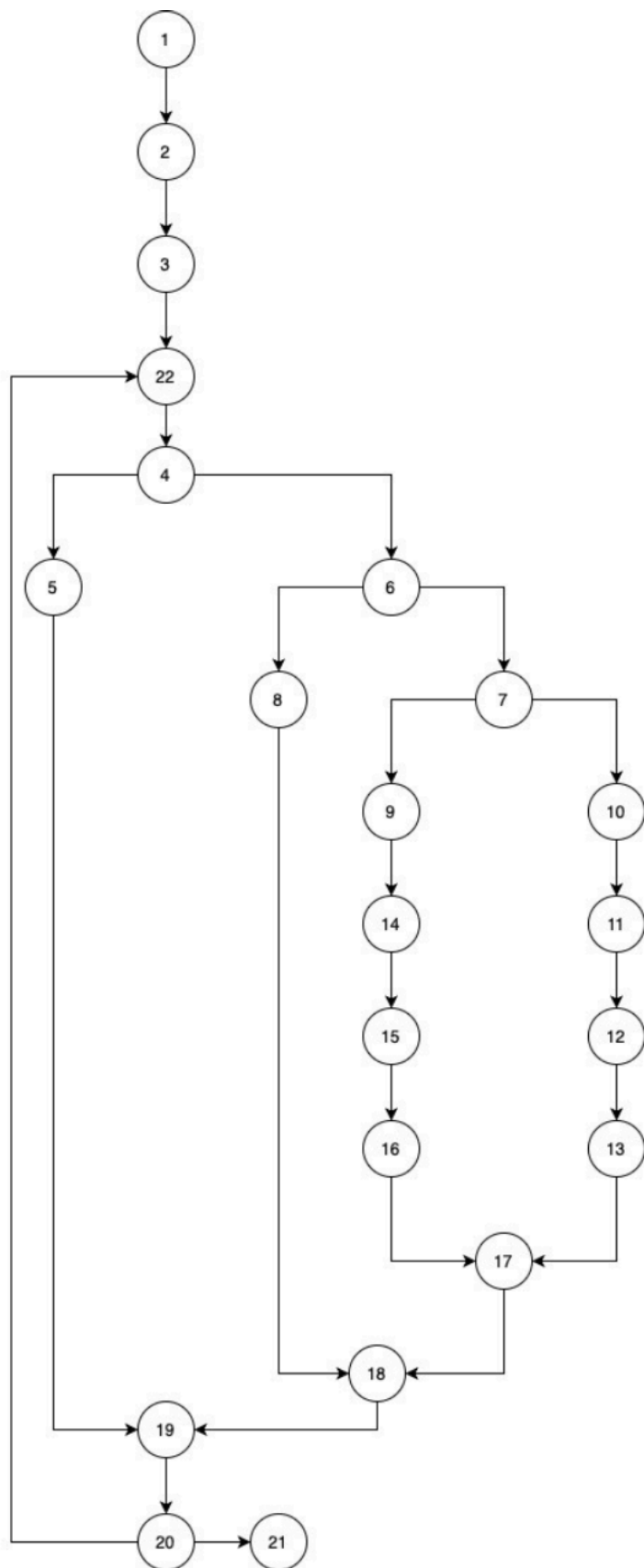
1.4. Графовая модель:



Исходный граф:



Модифицированный граф:



2.1. Оценивание структурной сложности с помощью критерия:
минимальное покрытие вершин и дуг графа управления:

$Y = 24$ – общее число дуг графа;
 $N = 22$ – общее число вершин графа;
 $\Omega = 1$ – число связных компонент
 $n_b = 4$ – число вершин, в которых происходит ветвление.
 $Z = Y - N + 2 * \Omega = 24 - 22 + 2 * 1 = 4$
 $Z = n_b + 1 = 4 + 1 = 5$
M1: 1-2-3-22-4-5-19-20-22-4-6-8-18-19-20-21
M2: 1-2-3-22-4-6-7-9-14-15-16-17-18-19-20-21
M3: 1-2-3-22-4-6-7-10-11-12-13-17-18-19-20-21
S = 13

2.2. Оценивание структурной сложности с помощью критерия: выбор маршрутов на основе цикломатического числа графа:

$$Z = n_b + 1 = 4 + 1 = 5$$

M1: 22-4-5-19-20
M2: 1-2-3-22-4-5-19-20-21
M3: 1-2-3-22-4-6-8-18-19-20-21
M4: 1-2-3-22-4-6-7-9-14-15-16-17-18-19-20-21
M5: 1-2-3-22-4-6-7-10-11-12-13-17-18-19-20-21

$$S = 15$$

2.3. Программный расчет:

```

Min ways....
----- Path #1 -----
-> 1 -> 2 -> 3 -> 22 -> 4 -> 5 -> 19 -> 20 -> 22 -> 4 -> 6 -> 7 -> 9 -> 14 -> 1
5 -> 16 -> 17 -> 18 -> 19 -> 20 -> 22 -> 4 -> 6 -> 8 -> 18 -> 19 -> 20 -> 22 ->
4 -> 6 -> 7 -> 10 -> 11 -> 12 -> 13 -> 17 -> 18 -> 19 -> 20 -> 21
-----Press a key to continue -----
Complexity = 13
Press a key...

Z ways....
----- Path #1 -----
-> 22 -> 4 -> 5 -> 19 -> 20 -> 22
-----Press a key to continue -----
----- Path #1 -----
-> 1 -> 2 -> 3 -> 22 -> 4 -> 5 -> 19 -> 20 -> 21
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 3 -> 22 -> 4 -> 6 -> 7 -> 9 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 ->
20 -> 21
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 3 -> 22 -> 4 -> 6 -> 7 -> 10 -> 11 -> 12 -> 13 -> 17 -> 18 -> 19 ->
20 -> 21
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 3 -> 22 -> 4 -> 6 -> 8 -> 18 -> 19 -> 20 -> 21
-----Press a key to continue -----
Complexity = 15

```

Вывод:

В результате выполнения работы была произведена оценка структурной сложности двух программ с помощью критериев: минимального покрытия дуг графа и выбора маршрутов на основе цикломатического числа графа. Расчеты были проведены как ручным, так и программным способами.