

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. 6304

Тимофеев А.А.

Преподаватель

Кирияничков В.А.

Санкт-Петербург

2020

Задание

Выполнить оценивание структурной сложности двух программ с помощью критериев:

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

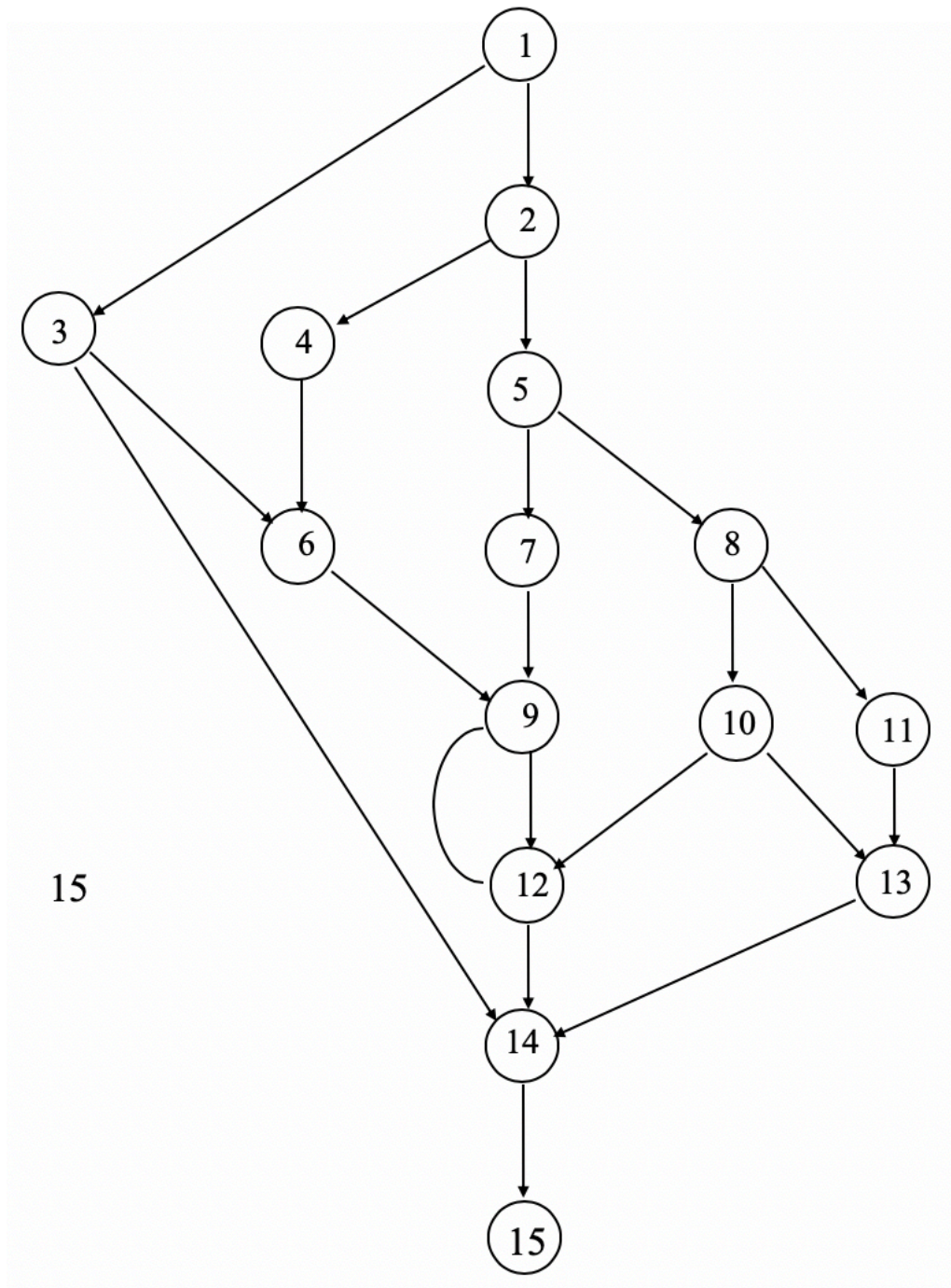
Варианты программ:

- 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.

Вариант 15.



Ход работы

1. Оценивание структурной сложности первой программы с помощью критерия минимального покрытия дуг графа.

1.1. Расчет сложности вручную.

Ветвления: 1, 2, 3, 5, 8, 10, 12.

Минимальный набор путей:

1. $\underline{1} - \underline{3} - 14 - 15 = 2$
2. $\underline{1} - \underline{3} - 6 - 9 - \underline{12} - 14 - 15 = 3$
3. $\underline{1} - \underline{2} - \underline{5} - \underline{8} - 11 - 13 - 14 - 15 = 4$
4. $\underline{1} - \underline{2} - \underline{5} - \underline{8} - \underline{10} - 13 - 14 - 15 = 5$
5. $\underline{1} - \underline{2} - \underline{5} - \underline{8} - \underline{10} - \underline{12} - 9 - \underline{12} - 14 - 15 = 7$
6. $\underline{1} - \underline{2} - 4 - 6 - 9 - \underline{12} - 14 - 15 = 3$
7. $\underline{1} - \underline{2} - \underline{5} - 7 - 9 - \underline{12} - 14 - 15 = 4$

Сложность равна 28

1.2. Расчет сложности при помощи программы ways.exe.

Граф для программы:

```
Nodes{
    1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
}
```

```
Top{1}
```

```
Last{15}
```

```
Arcs{
    arc(1,2);
    arc(1,3);
    arc(2,4);
    arc(2,5);
    arc(3,6);
    arc(3,14);
    arc(4,6);
    arc(5,7);
    arc(5,8);
    arc(6,9);
    arc(7,9);
    arc(8,10);
    arc(8,11);
    arc(9,12);
    arc(10,12);
    arc(10,13);
```

```

arc(11,13);
arc(12,9);
arc(12,14);
arc(13,14);
arc(14,15);
}

```

Минимальный набор путей:

1. 1 – 2 – 4 – 6 – 9 – 12 – 9 – 12 – 14 – 15
2. 1 – 3 – 6 – 9 – 12 – 14 – 15
3. 1 – 2 – 5 – 7 – 9 – 12 – 14 – 15
4. 1 – 2 – 5 – 8 – 10 – 12 – 14 – 15
5. 1 – 2 – 5 – 8 – 11 – 13 – 14 – 15
6. 1 – 2 – 5 – 8 – 10 – 13 – 14 – 15
7. 1 – 3 – 14 – 15

Сложность равна 28.

1.3. Сравнение результатов

Программный результат от ручного отличается выбором пути для прохождения цикла в нем. Получившиеся сложности одинаковы в обоих вариантах.

2. Оценивание структурной сложности первой программы с помощью критерия на основе цикломатического числа.

2.1. Расчет сложности вручную.

Количество рёбер – 21.

Количество вершин – 15.

Для связного графа требуется добавить 1 ребро из вершины 15 в вершину 1.

Цикломатическое число равно $= 21 - 15 + 2 * 1 = 8$.

Ветвления: 1, 2, 3, 5, 8, 10, 12.

Набор путей:

1. 1 – 2 – 4 – 6 – 9 – 12 – 14 – 15 = 3
2. 1 – 2 – 5 – 7 – 9 – 12 – 14 – 15 = 4
3. 1 – 2 – 5 – 8 – 10 – 12 – 14 – 15 = 6
4. 1 – 2 – 5 – 8 – 10 – 13 – 14 – 15 = 5
5. 1 – 2 – 5 – 8 – 11 – 13 – 14 – 15 = 4
6. 1 – 3 – 6 – 9 – 12 – 14 – 15 = 3
7. 1 – 3 – 14 – 15 = 2

8. $9 - 12 - 9 = 1$

Сложность равна 28.

2.2. Расчет сложности при помощи программы ways.exe.

Пути:

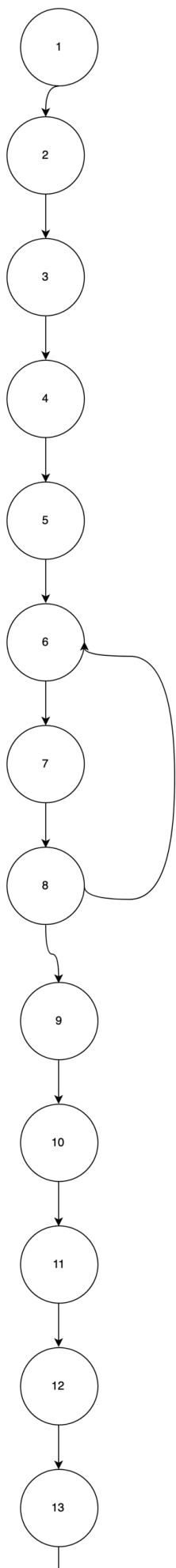
1. $1 - 2 - 4 - 6 - 9 - 12 - 14 - 15$
2. $1 - 2 - 5 - 7 - 9 - 12 - 14 - 15$
3. $1 - 2 - 5 - 8 - 10 - 12 - 14 - 15$
4. $1 - 2 - 5 - 8 - 10 - 13 - 14 - 15$
5. $1 - 2 - 5 - 5 - 11 - 13 - 14 - 15$
6. $1 - 3 - 6 - 9 - 12 - 14 - 15$
7. $1 - 3 - 14 - 15$
8. $9 - 12 - 9$

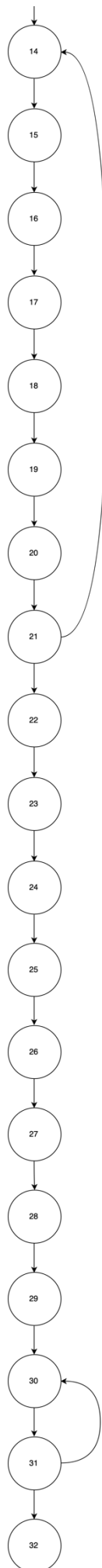
Сложность равна 28.

2.3. Сравнение результатов.

Пути и сложность идентичны.

3. Оценивание структурной сложности второй программы (из л/р 1) с помощью критерия минимального покрытия дуг графа. Код программы на Си представлен в приложении А.





3.1. Вручную

Ветвления: 8, 21, 31.

Минимальный набор путей:

$1 - 2 - 3 - 4 - 5 - 6 - 7 - \underline{8} - 6 - 7 - \underline{8} - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - \underline{21} - 14 - 15 - 16 - 17 - 18 - 19 - 20 - \underline{21} - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 - \underline{31} - 30 - \underline{31} - 32 = 6$

Сложность равна 6.

3.2. С помощью программы ways.exe

Граф для программы:

Nodes{

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32

}

Top{1}

Last{32}

Arcs{

arc(1, 2);

arc(2, 3);

arc(3, 4);

arc(4, 5);

arc(5, 6);

arc(6, 7);

arc(7, 8);

arc(8, 6);

arc(8, 9);

arc(9, 10);

arc(10, 11);

arc(11, 12);

arc(12, 13);

arc(13, 14);

arc(14, 15);

arc(15, 16);

```

arc(16,17);
arc(17,18);
arc(18,19);
arc(19,20);
arc(20,21);
arc(21,14);
arc(21,22);
arc(22,23);
arc(23,24);
arc(24,25);
arc(25,26);
arc(26,27);
arc(27,28);
arc(28,29);
arc(29,30);
arc(30,31);
arc(31,30);
arc(31,32);
}

```

Минимальный набор путей:

1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 –
 17 – 18 – 19 – 20 – 21 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 –
 25 – 26 – 27 – 28 – 29 – 30 – 31 – 30 – 31 – 32

Сложность равна 6.

3.3. Сравнение результатов.

Пути и сложности идентичны.

4. Оценивание структурной сложности второй программы (из л/р 1) с помощью критерия на основе цикломатического числа.

4.1. Расчет сложности вручную.

Количество рёбер – 34.

Количество вершин – 32.

Для связного графа требуется добавить 1 ребро из вершины №32 в вершину №1.

Цикломатическое число равно $= 34 - 32 + 2 \cdot 1 = 4$.

Ветвления: 8, 21, 31.

Набор путей:

1. $6 - 7 - \underline{8} - 6$
2. $14 - 15 - 16 - 17 - 18 - 19 - 20 - \underline{21} - 14$
3. $30 - \underline{31} - 30$
4. $1 - 2 - 3 - 4 - 5 - 6 - 7 - \underline{8} - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - \underline{21} - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 - \underline{31} - 32$

Сложность равна 6.

4.2. Расчет сложности при помощи программы ways.exe.

Пути:

1. $6 - 7 - \underline{8} - 6$
2. $14 - 15 - 16 - 17 - 18 - 19 - 20 - \underline{21} - 14$
3. $30 - \underline{31} - 30$
4. $1 - 2 - 3 - 4 - 5 - 6 - 7 - \underline{8} - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - \underline{21} - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 - \underline{31} - 32$

Сложность равна 6.

4.3. Сравнение результатов.

Пути и сложности идентичны.

Выводы

В ходе выполнения данной лабораторной работы были получены практические навыки оценивания структурной сложности программ. Была проведена оценка структурной сложности двух программ.

ПРИЛОЖЕНИЕ А

Код программы из первой лабораторной работы

```
#include <math.h>
#include <time.h>
#include <stdlib.h>

#define N 100

void linfit2(float *x, float *y, float *y_calc, float *a, float *b, int n) {
    float sum_x = 0.0;
    float sum_y = 0.0;
    float sum_xy = 0.0;
    float sum_x2 = 0.0;
    float sum_y2 = 0.0;
    for (int i = 0; i < N; i++) {
        float xi = x[i];
        float yi = y[i];
        sum_x += xi;
        sum_y += yi;
        sum_xy += (xi * yi);
        sum_x2 += (xi * xi);
        sum_y2 += (yi * yi);
    }

    float sxx = sum_x2 - sum_x * sum_x / n;
    float sxy = sum_xy - sum_x * sum_y / n;
    float syy = sum_y2 - sum_y * sum_y / n;

    (*b) = sxy / sxx;
    (*a) = ( (sum_x2 * sum_y - sum_x * sum_xy) / n ) / sxx;
    float correl_coef = sxy / sqrt(sxx * syy);
    float see = sqrt( (sum_y2 - *a * sum_y - *b * sum_xy) / (n - 2) );
    float sigma_b = see / sqrt(sxx);
    float sigma_a = sigma_b * sqrt(sum_x2 / n);
    for (int i = 0; i < n; i++) {
        y_calc[i] = *a + *b * x[i];
    }
}

int main() {
```

```
float x[N] = {0.0};
float y[N] = {0.0};

float y_calc[N] = {0.0};

float a = 0.0;
float b = 0.0;

srand(time(NULL));

for (int i = 0; i < N; i++) {
    x[i] = rand() % 100;
    y[i] = rand() % 100;
}

linfit2(x, y, y_calc, &a, &b, N);

return 0;
}
```