

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. 7304

Комаров А.О.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Задание

Выполнить оценивание структурной сложности двух программ с помощью критериев:

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

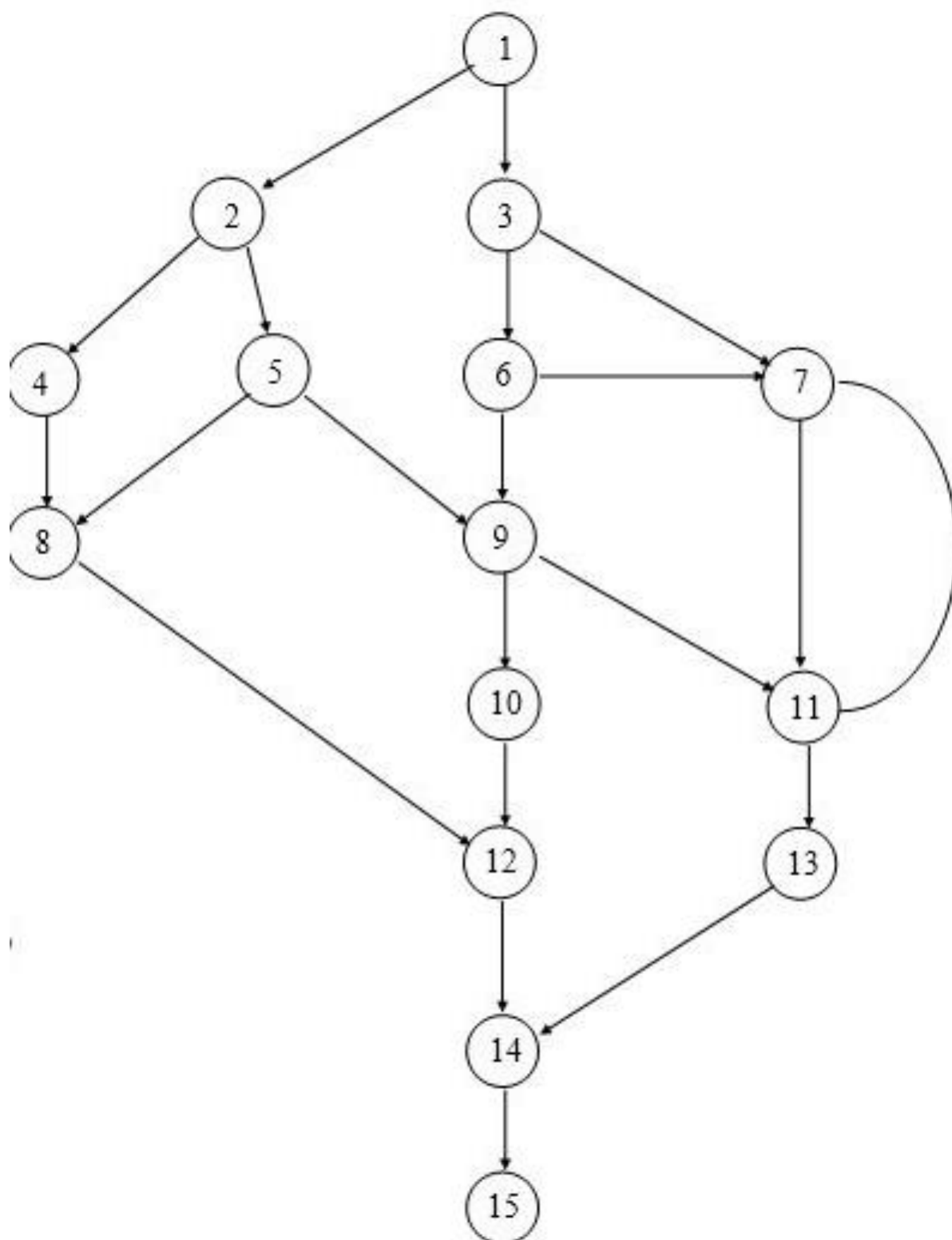
Варианты программ:

- 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 7.



Ход работы

1.1. Программа из задания

1.1.1. Критерий минимального покрытия

Маршруты для минимального покрытия:

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

Сложность программы:

$$S_2 = \sum_{i=1}^M \xi_i = 2 + 3 + 6 + 4 + 3 + 5 = 23. \quad (3.1)$$

1.1.2. Анализ базовых маршрутов

Число вершин графа $N = 15$, число дуг графа $Y = 21$, число связных компонент $\Omega = 1$. Цикломатическое число Z :

$$Z = Y - N + 2 * \Omega = 8. \quad (3.2)$$

Маршруты:

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

Сложность программы:

$$S_2 = \sum_{i=1}^M \xi_i = 2 + 2 + 3 + 4 + 5 + 4 + 3 + 4 = 27. \quad (3.3)$$

1.1.3. Программный анализ

Граф задан в нотации приложенной программы. Файл с описанием — в приложении А.

С графом в приведенном описании возникла ошибка работы из-за ребра 9–11. Для решения добавлено дополнительная вершина. Лог работы программы в приложении Б.

Маршруты для минимального покрытия:

1. 1, 2, 4, 8, 12, 14, 15;
2. 1, 2, 5, 8, 12, 14, 15;
3. 1, 2, 5, 9, 10, 12, 14, 15;
4. 1, 2, 5, 9, 16, 13, 14, 15;
5. 1, 3, 6, 7, 11, 7, 11, 16, 13, 14, 15;
6. 1, 3, 6, 9, 10, 12, 14;
7. 1, 3, 7, 11, 16, 13, 14, 15.

Сложность: 25

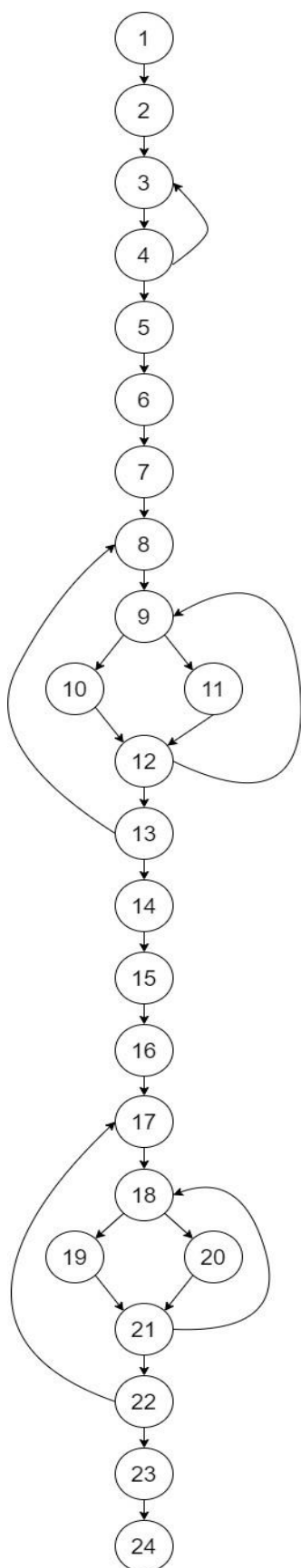
Базовые

маршруты:

1. 7, 11, 7;
2. 1, 2, 4, 8, 12, 14, 15;
3. 1, 2, 5, 8, 12, 14, 15;
4. 1, 2, 5, 9, 10, 12, 14, 15;
5. 1, 2, 5, 9, 16, 13, 14, 15;
6. 1, 3, 6, 7, 11, 16, 13, 14, 15;
7. 1, 3, 6, 9, 10, 12, 14, 15;
8. 1, 3, 7, 11, 16, 13, 14, 15.

Сложность: 25

2.1 Оценивание структурной сложности второй программы (из л/р 1) с помощью критерия минимального покрытия дуг графа



2.1.1 Вручную

Ветвления: 4,9,12,13,18,21,22.

Минимальный набор путей:

1) 1-2-3-4-3-4-5-6-7-8-9-10-12-9-11-12-13-8-9-10-12-13-14-15-16-17-18-19-21-18-20-21-22-17-18-19-21-22-23-24 (18 ветвлений)

Сложность равна 18.

2.1.2 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}
Top{1}
Last{24
} Arcs{
arc(1,2);
arc(2,3);
arc(3,4);
arc(4,5);
arc(4,3);
arc(5,6);
arc(6,7);
arc(7,8);
arc(8,9);
arc(9,10);
arc(9,11);
arc(10,12);
arc(11,12);
arc(12,13);
arc(12,9);
arc(13,14);
arc(13,8);
arc(14,15);
arc(15,16);
arc(16,17);
arc(17,18);
arc(18,19);
arc(18,20);
arc(19,21);
arc(20,21);
arc(21,22);
arc(21,18);
arc(22,23);
arc(22,17);
arc(23,24);
}
```

Минимальный набор путей:

1) 1-2-3-4-3-4-5-6-7-8-9-10-12-9-11-12-13-8-9-10-12-13-14-15-16-17-18-19-21-18-20-21-22-17-18-19-21-22-23-24

Сложность равна 18.

3.3. Сравнение результатов.

Все пути и сложность совпадают.

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

2.2.1 Вручную

Количество рёбер – 30.

Количество вершин – 24.

Для связного графа требуется добавить 1 ребро из вершины №24 в вершину №1.

Цикломатическое число равно $= 30 - 24 + 2 \cdot 1 = 8$.

Ветвления: 4,9,12,13,18,21,22.

Набор путей:

- 1) 3-~~4~~-3 (1 ветвление)
- 2) 8-~~9~~-10-~~12~~-~~13~~-8 (3 ветвления)
- 3) ~~9~~-10-~~12~~-~~9~~ (3 ветвления)
- 4) 17-~~18~~-19-~~21~~-~~22~~-17 (3 ветвления)
- 5) ~~18~~-19-~~21~~-~~18~~ (3 ветвления)
- 6) 1-2-3-~~4~~-5-6-7-8-~~9~~-11-~~12~~-~~13~~-14-15-16-17-~~18~~-19-~~21~~-~~22~~-23-24 (7 ветвлений)
- 7) 1-2-3-~~4~~-5-6-7-8-~~9~~-10-~~12~~-~~13~~-14-15-16-17-~~18~~-19-~~21~~-~~22~~-23-24 (7 ветвлений)
- 8) 1-2-3-~~4~~-5-6-7-8-~~9~~-10-~~12~~-~~13~~-14-15-16-17-~~18~~-20-~~21~~-~~22~~-23-24 (7 ветвлений)

Сложность равна 34.

2.2.2 С помощью программы

ways.exe.Пути:

- 1) 3-4-3
- 2) 9-10-12-9
- 3) 8-9-10-12-13-8
- 4) 18-19-21-18
- 5) 17-18-19-21-22-17
- 6) 1-2-3-4-5-6-7-8-9-10-12-13-14-15-16-17-18-19-21-22-23-24
- 7) 1-2-3-4-5-6-7-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24
- 8) 1-2-3-4-5-6-7-8-9-11-12-13-14-15-16-17-18-19-21-22-23-24

Сложность равна 34.

2.2.3 Сравнение результатов.

Все пути и сложность совпадают.

Выводы

В результате выполнения данной лабораторной работы были изучены критерии оценивания структурной сложности программ. Была проведена оценка структурной сложности двух программ: соответствующая варианту и из первой лабораторной работы.

ПРИЛОЖЕНИЕ А

Граф из задания

```
1  Nodes { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 }
2
3  Top { 1 }
4
5  Last { 15 }
6
7  Arcs {
8      arc(1, 2);
9      arc(1, 3);
10     arc(2, 4);
11     arc(2, 5);
12     arc(3, 6);
13     arc(3, 7);
14     arc(4, 8);
15     arc(5, 8);
16     arc(5, 9);
17     arc(6, 7);
18     arc(6, 9);
19     arc(7, 11);
20     arc(8, 12);
21     arc(9, 10);
22     arc(9, 16);
23     arc(10, 12);
24     arc(11, 7);
25     arc(11, 16);
26     arc(12, 14);
27     arc(13, 14);
28     arc(14, 15);
29     arc(16, 13);
30 }
```

ПРИЛОЖЕНИЕ Б

логгирование графа из задания

```
1  Min ways....
2  ..... Path #1 .....
3  -> 1 -> 2 -> 4 -> 8 -> 12 -> 14 -> 15
4  -----Press a key to continue -----
5  ..... Path #2 .....
6  -> 1 -> 3 -> 6 -> 7 -> 11 -> 7 -> 11 -> 16 -> 13 -> 14 -> 15
7  -----Press a key to continue -----
8  ..... Path #3 .....
9  -> 1 -> 2 -> 5 -> 8 -> 12 -> 14 -> 15
10 -----Press a key to continue -----
11 ..... Path #4 .....
12 12 -> 1 -> 2 -> 5 -> 9 -> 10 -> 12 -> 14 -> 15
13 -----Press a key to continue -----
14 ..... Path #5 .....
15 15 -> 1 -> 2 -> 5 -> 9 -> 16 -> 13 -> 14 -> 15
16 -----Press a key to continue -----
17 ..... Path #6 .....
18 18 -> 1 -> 3 -> 7 -> 11 -> 16 -> 13 -> 14 -> 15
19 -----Press a key to continue -----
20 ..... Path #7 .....
21 21 -> 1 -> 3 -> 6 -> 9 -> 10 -> 12 -> 14 -> 15
22 -----Press a key to continue -----
23
24 Complexity = 25
25 Press a key...
26
27 Z ways....
28 ..... Path #1 .....
29 29 -> 7 -> 11 -> 7
30 -----Press a key to continue -----
31 ..... Path #1 .....
32 32 -> 1 -> 2 -> 4 -> 8 -> 12 -> 14 -> 15
```

33 -----Press a key to continue -----

34 Path #2

35 -> 1 -> 2 -> 5 -> 8 -> 12 -> 14 -> 15

36 -----Press a key to continue -----

37 Path #3

38 -> 1 -> 2 -> 5 -> 9 -> 10 -> 12 -> 14 -> 15

39 -----Press a key to continue -----

40 Path #4

41 -> 1 -> 2 -> 5 -> 9 -> 16 -> 13 -> 14 -> 15

42 -----Press a key to continue -----

43 Path #5

44 -> 1 -> 3 -> 6 -> 7 -> 11 -> 16 -> 13 -> 14 -> 15

45 -----Press a key to continue -----

46 Path #6

47 -> 1 -> 3 -> 6 -> 9 -> 10 -> 12 -> 14 -> 15

48 -----Press a key to continue -----

49 Path #7

50 -> 1 -> 3 -> 7 -> 11 -> 16 -> 13 -> 14 -> 15

51 -----Press a key to continue -----

52

53 Complexity = 25

54 Press a key...

ПРИЛОЖЕНИЕ В

Граф ЛР1 в нотации программы-анализатора

```
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}  
Top{1}  
Last{24  
}  
Arcs{  
  arc(1,2);  
  arc(2,3);  
  arc(3,4);  
  arc(4,5);  
  arc(4,3);  
  arc(5,6);  
  arc(6,7);  
  arc(7,8);  
  arc(8,9);  
  arc(9,10);  
  arc(9,11);  
  arc(10,12);  
  arc(11,12);  
  arc(12,13);  
  arc(12,9);  
  arc(13,14);  
  arc(13,8);  
  arc(14,15);  
  arc(15,16);  
  arc(16,17);  
  arc(17,18);  
  arc(18,19);  
  arc(18,20);  
  arc(19,21);  
  arc(20,21);  
  arc(21,22);  
  arc(21,18);  
  arc(22,23);  
  arc(22,17);  
  arc(23,24);  
}
```