

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

Нгуен Т.Т.З.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучены методы оценки структурной сложности графовых моделей программ.

Постановка задачи.

1. Выполнить оценивание структурной сложности **двух программ** с помощью критериев:
 - Минимального покрытия вершин и дуг графа управления;
 - Выбора маршрутов на основе цикломатического числа графа.
2. 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).
3. Оцениваемые характеристики структурной сложности:
 - Число учитываемых маршрутов проверки программы для заданного критерия;
 - Цикломатическое число;
 - Суммарное число ветвлений по всем маршрутам – оценка структурной сложности;

Ход выполнения.

Вариант №11:

1. Программа с заданной структурой управляющего графа, структура управляющего графа, выбираемой из файла `zadan_struct.doc`

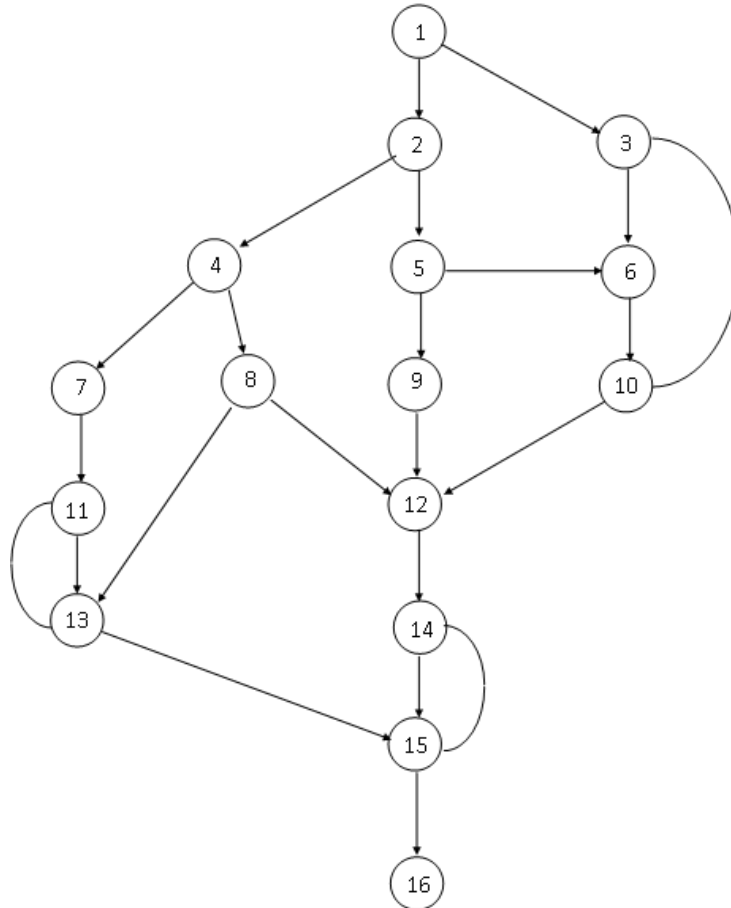


Рисунок 1: Управляющий граф программы из файла `zadan_struct.doc` для определения структурной сложности

1.1. Анализ структурной сложности графа (без использования программы `ways.exe`).

В соответствии с первым критерием надо выбрать минимальное количество маршрутов с обходом всех вершин и дуг графа по возможности без повторных проходов (без повторов).

M1: 1 – 2 – 5 – 9 – 12 – 14 – 15 – 14 – 15 – 16 ;

M2: 1 – 2 – 5 – 6 – 10 – 12 – 14 – 15 – 16 ;

M3: 1 – 3 – 6 – 10 – 3 – 6 – 10 – 12 – 14 – 15 – 16 ;

M4: 1 – 2 – 4 – 7 – 11 – 13 – 11 – 13 – 15 – 16 ;

M5: 1 – 2 – 4 – 8 – 13 – 15 – 16 ;

М6: 1 – 2 – 4 – 8 – 12 – 14 – 15 – 16.

т.е. по первому критерию требуется 6 маршрута $M = 6$.

1.1.1. Сложность программы по первому критерию вычисляется следующим образом:

М1: 1 – 2 – 5 – 9 – 12 – 14 – 15 – 14 – 15 – 16 ; = 5

М2: 1 – 2 – 5 – 6 – 10 – 12 – 14 – 15 – 16 ; = 5

М3: 1 – 3 – 6 – 10 – 3 – 6 – 10 – 12 – 14 – 15 – 16 ; = 4

М4: 1 – 2 – 4 – 7 – 11 – 13 – 11 – 13 – 15 – 16; = 6

М5: 1 – 2 – 4 – 8 – 13 – 15 – 16; = 6

М6: 1 – 2 – 4 – 8 – 12 – 14 – 15 – 16. = 5

Сложность программы: $S = 5 + 5 + 4 + 6 + 6 + 5 = 31$ единиц

1.1.2. Расчет структурной сложности программы вручную по первому критерию.

Второй критерий рассматривает все маршруты, отличающиеся хотя бы одной дугой. Для рассматриваемого нами графа $N = 16$ - полное число вершин, $Y = 23$ - количество связывающих дуг и $P = 1$ - число связных компонент

$$Z = Y - N + 2 \times P = 23 - 16 + 2 \times 1 = 9$$

то есть цикломатическое число равно 9.

1.1.3. Построить 9 линейно-независимых циклов и линейно-независимых маршрутов, после чего была подсчитана структурная сложность по второму критерию:

М1: 14 – 15;

М2: 11 – 13;

М3: 3 – 6 – 10;

М4: 1 – 2 – 5 – 9 – 12 – 14 – 15 – 16;

М5: 1 – 2 – 5 – 6 – 10 – 12 – 14 – 15 – 16;

М6: 1 – 2 – 4 – 8 – 12 – 14 – 15 – 16;

М7: 1 – 2 – 4 – 8 – 13 – 15 – 16;

М8: 1 – 2 – 4 – 7 – 11 – 13 – 14 – 15 – 16;

М9: 1 – 3 – 6 – 10 – 12 – 14 – 15 – 16;

Подсчитана структурная сложность

M1: 14 – <u>15</u> ;	= 1
M2: 11 – <u>13</u> ;	= 1
M3: 3 – 6 – <u>10</u> ;	= 1
M4: <u>1</u> – <u>2</u> – <u>5</u> – 9 – 12 – 14 – <u>15</u> – 16;	= 4
M5: <u>1</u> – <u>2</u> – <u>5</u> – 6 – <u>10</u> – 12 – 14 – <u>15</u> – 16;	= 5
M6: <u>1</u> – <u>2</u> – <u>4</u> – <u>8</u> – 12 – 14 – <u>15</u> – 16;	= 5
M7: <u>1</u> – <u>2</u> – <u>4</u> – <u>8</u> – <u>13</u> – <u>15</u> – 16;	= 6
M8: <u>1</u> – <u>2</u> – <u>4</u> – 7 – 11 – <u>13</u> – 14 – <u>15</u> – 16;	= 5
M9: <u>1</u> – 3 – 6 – <u>10</u> – 12 – 14 – <u>15</u> – 16;	= 3

Сложность программы $S = 1+1+1+4+5+5+6+5+3 = 31$ единиц.

1.2. Анализ структурной сложности графа (использования программы *ways.exe*).

Описание графа на входном языке программы анализа имеет вид:

```
Nodes{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
```

```
Top{1}
```

```
Last{16}
```

```
Arcs{
```

```
arc(1, 2);
```

```
arc(1, 3);
```

```
arc(2, 4);
```

```
arc(2, 5);
```

```
arc(3, 6);
```

```
arc(4, 7);
```

```
arc(4, 8);
```

```
arc(5, 6);
```

```
arc(5, 9);
```

```
arc(6, 10);
```

```
arc(7, 11);
```

```
arc(8, 12);
```

```
arc(8, 13);
```

```
arc(9, 12);
```

```
arc(10, 12);
```

```
arc(10, 3);
```

```
arc(11, 13);
```

```
arc(12, 14);
```

```
arc(13, 11);
```

```
arc(13, 15);
```

```
arc(14, 15);
```

```
arc(15, 14);
```

```
arc(15, 16);
```

```
}
```

Результаты анализа графа с помощью программы *ways.exe* следующие:

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: WAYS
Min ways....
----- Path #1 -----
-> 1 -> 2 -> 4 -> 7 -> 11 -> 13 -> 11 -> 13 -> 15 -> 14 -> 15 -> 16
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 3 -> 6 -> 10 -> 3 -> 6 -> 10 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 5 -> 6 -> 10 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 4 -> 8 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----
----- Path #5 -----
-> 1 -> 2 -> 4 -> 8 -> 13 -> 15 -> 16
-----Press a key to continue -----
----- Path #6 -----
-> 1 -> 2 -> 5 -> 9 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----

Complexity = 31
Press a key...
_
```

Рисунок 2: Программный расчёт структурной сложности первой программы по первому критерию

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: WAYS
Z ways....
----- Path #1 -----
-> 3 -> 6 -> 10 -> 3
-----Press a key to continue -----
----- Path #2 -----
-> 11 -> 13 -> 11
-----Press a key to continue -----
----- Path #3 -----
-> 14 -> 15 -> 14
-----Press a key to continue -----
----- Path #1 -----
-> 1 -> 2 -> 4 -> 7 -> 11 -> 13 -> 15 -> 16
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 4 -> 8 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 4 -> 8 -> 13 -> 15 -> 16
-----Press a key to continue -----
----- Path #4 -----
-> 1 -> 2 -> 5 -> 6 -> 10 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----
----- Path #5 -----
-> 1 -> 2 -> 5 -> 9 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----
----- Path #6 -----
-> 1 -> 3 -> 6 -> 10 -> 12 -> 14 -> 15 -> 16
-----Press a key to continue -----

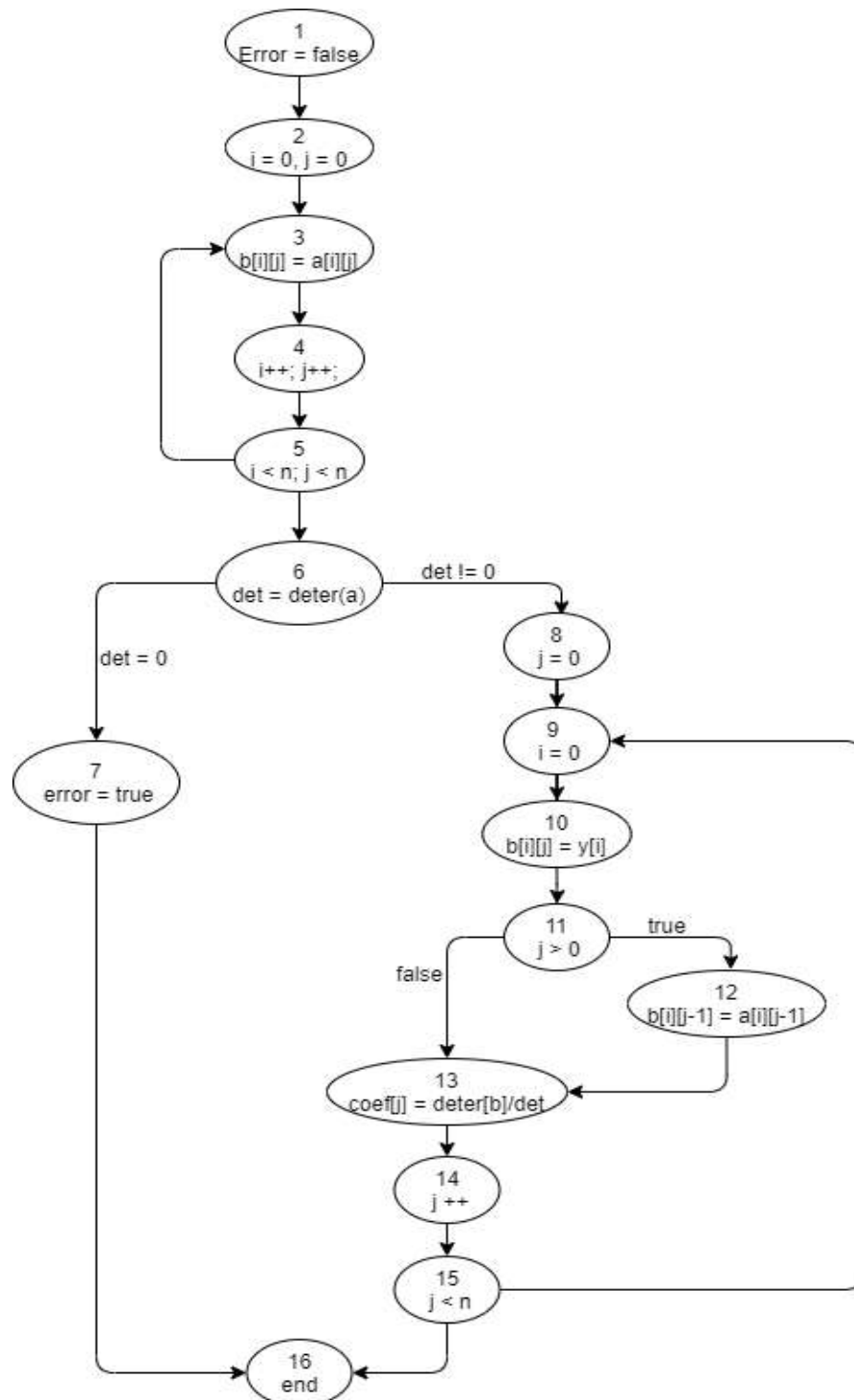
Complexity = 31
Press a key...
```

Рисунок 3: Программный расчёт структурной сложности первой программы по второму критерию

2. Программы из 1-ой лабораторной работы:

2.1. Анализ структурной сложности графа (без использования программы *ways.exe*).

Был составлен управляющий граф для программы первого лабораторной работы и представлен на Рисунке 4:



Рисунке 4: Управляющий граф программы из первой лабораторной работы для определения структурной сложности

2.1.1. Расчет структурной сложности программы из первой лабораторной работы вручную по первому критерию.

$$M1: 1 - 2 - 3 - 4 - \underline{5} - 3 - 4 - \underline{5} - \underline{6} - 7 - 16; \quad = 3$$

$$M2: 1 - 2 - 3 - 4 - \underline{5} - \underline{6} - 8 - 9 - 10 - \underline{11} - 12 - 13 -$$

$$14 - \underline{15} - 9 - 10 - \underline{11} - 13 - 14 - \underline{15} - 16;$$

Сложность программы $S = 3 + 6 = 9$ единиц.

2.1.2. Расчет структурной сложности программы вручную по первому критерию

Для рассматриваемого нами графа $N = 16$ - полное число вершин, $Y = 19$ - количество связывающих дуг и $P = 1$ - число связных компонент и цикломатическое число

$$Z = Y - N + 2 \times P = 19 - 16 + 2 \times 1 = 5$$

2.1.3. Построить 5 линейно-независимых циклов и линейно-независимых маршрутов, после чего была подсчитана структурная сложность по второму критерию:

$$M1: 3 - 4 - \underline{5}; \quad = 1$$

$$M2: 9 - 10 - \underline{11} - 13 - 14 - \underline{15}; \quad = 2$$

$$M3: 1 - 2 - 3 - 4 - \underline{5} - \underline{6} - 7 - 16; \quad = 2$$

$$M4: 1 - 2 - 3 - 4 - \underline{5} - \underline{6} - 8 - 9 - 10 - \underline{11} - 12 - 13 - 14 - \underline{15} - 16; \quad = 4$$

$$M5: 1 - 2 - 3 - 4 - \underline{5} - \underline{6} - 8 - 9 - 10 - \underline{11} - 13 - 14 - \underline{15} - 16 \quad = 4$$

Сложность программы $S = 13$ единицы

2.2. Анализ структурной сложности графа (использования программы ways.exe).

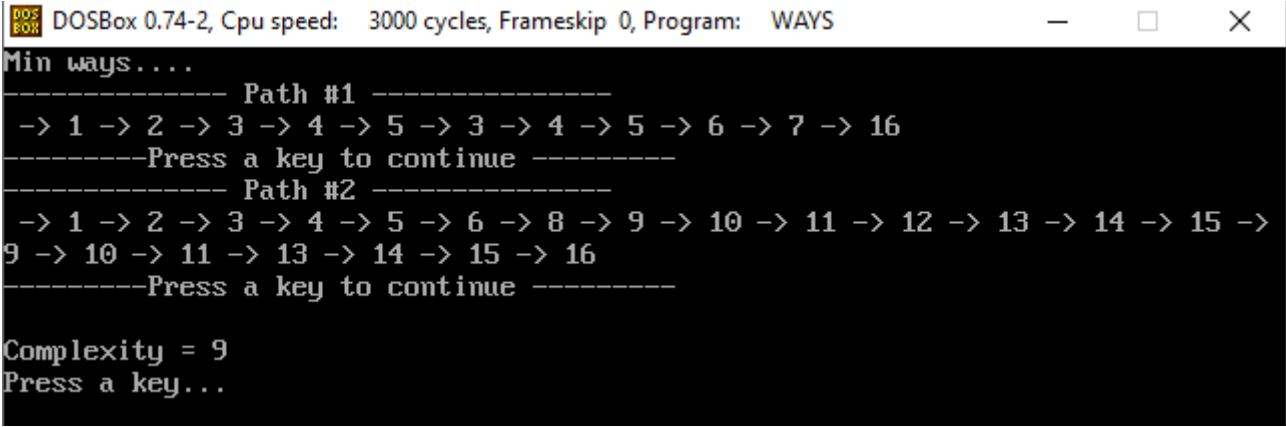
Описание графа на входном языке программы анализа имеет вид:

```
Nodes{
  1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
}

Top{1}
Last{16}

Arcs{
  arc(1, 2);
  arc(2, 3);
  arc(3, 4);
  arc(4, 5);
  arc(5, 3);
  arc(5, 6);
  arc(6, 7);
  arc(6, 8);
  arc(7, 16);
  arc(8, 9);
  arc(9, 10);
  arc(10, 11);
  arc(11, 12);
  arc(11, 13);
  arc(12, 13);
  arc(13, 14);
  arc(14, 15);
  arc(15, 9);
  arc(15, 16);
}
```

Результаты анализа графа с помощью программы ways.exe следующие:



```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: WAYS
Min ways....
----- Path #1 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 3 -> 4 -> 5 -> 6 -> 7 -> 16
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 8 -> 9 -> 10 -> 11 -> 12 -> 13 -> 14 -> 15 ->
9 -> 10 -> 11 -> 13 -> 14 -> 15 -> 16
-----Press a key to continue -----
Complexity = 9
Press a key...
```

Рисунок 5: Программный расчёт структурной сложности программы из первой лабораторной работы по первому критерию

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: WAYS
Z ways....
----- Path #1 -----
-> 3 -> 4 -> 5 -> 3
-----Press a key to continue -----
----- Path #2 -----
-> 9 -> 10 -> 11 -> 13 -> 14 -> 15 -> 9
-----Press a key to continue -----
----- Path #1 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 16
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 8 -> 9 -> 10 -> 11 -> 12 -> 13 -> 14 -> 15 ->
16
-----Press a key to continue -----
----- Path #3 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 8 -> 9 -> 10 -> 11 -> 13 -> 14 -> 15 -> 16
-----Press a key to continue -----
Complexity = 13
Press a key...
_
```

Рисунок 6: Программный расчёт структурной сложности программы из первой лабораторной работы по второму критерию

Выводы.

В ходе выполнения лабораторной работы были изучены методы оценки структурной сложности программы на основе его управляющего графа, была рассчитана структурная сложность двух программ с помощью двух критериев:

- минимального покрытия вершин и дуг графа управления;
- выбора маршрутов на основе цикломатического числа графа.

При этом были получены результаты расчетов, совпадающие ручным и программным.