

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

Виноградов К.А.

Преподаватель

Кирияничиков В.А.

Санкт-Петербург

2020

Цель работы.

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.

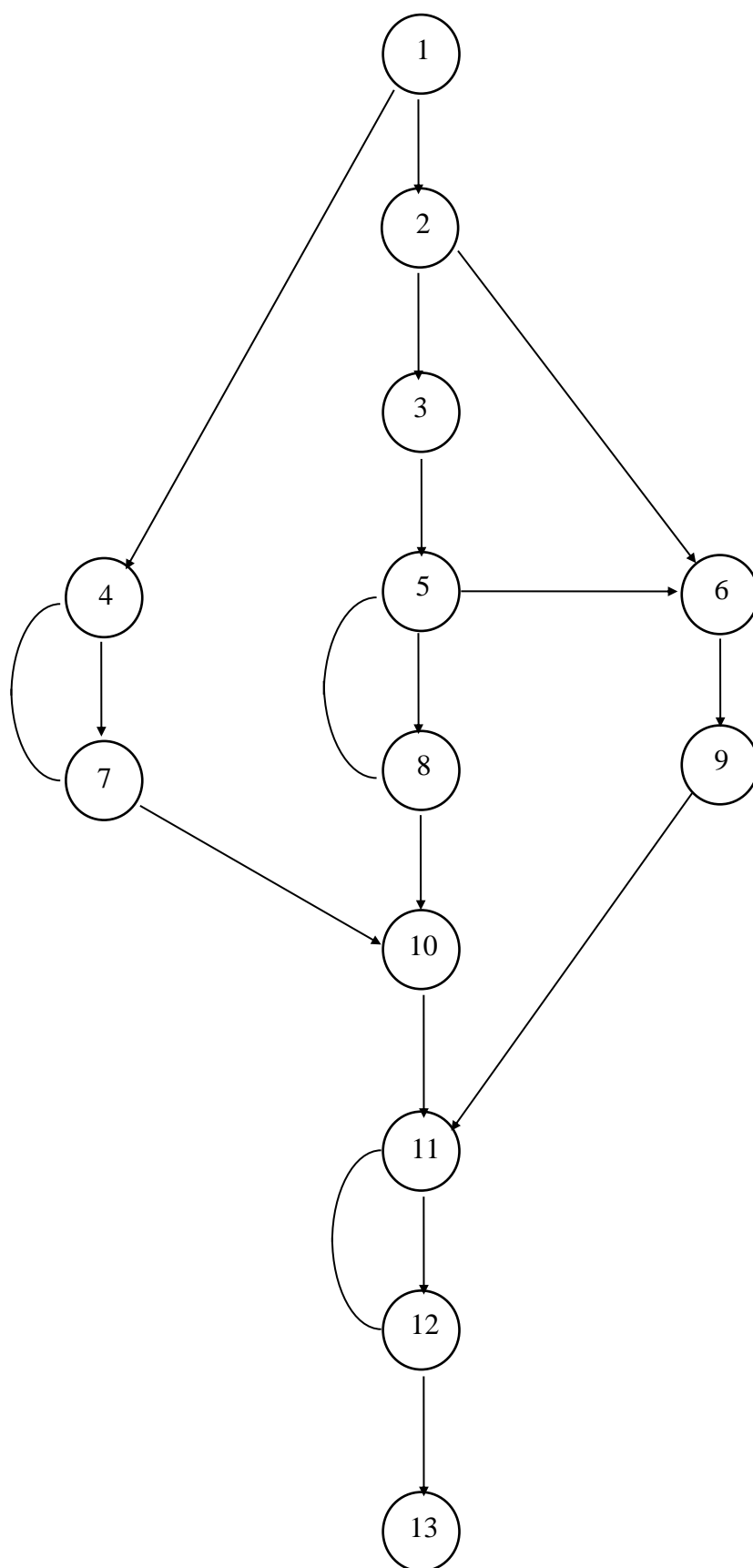


Рисунок 1 – Граф управления по заданию

Ход работы.

Оценим структурную сложность первой программы с помощью критерия минимального покрытия дуг графа. Сначала вручную.

Прежде всего подсчитаем число ветвлений – 1, 2, 5, 7, 8, 12. Всего 6.

Далее минимальный набор путей (жирным выделены узлы с ветвлениями):

- 1-4-7-4-7-10-11-12-11-12-13 (5 ветвлений)
- 1-2-3-5-8-5-8-10-11-12-13 (7 ветвлений)
- 1-2-3-5-6-9-11-12-13 (4 ветвления)
- 1-2-6-9-11-12-13 (3 ветвления)

Итого сложность равна: $5 + 7 + 4 + 3 = 19$.

Далее оценим структурную сложность с помощью критерия на основе цикломатического числа. Сначала вручную.

Число вершин в графе – 13, число ребер – 18. Для того, чтобы граф стал связным (из каждой вершины существовал путь в любую другую) достаточно добавить одно ребро из 13 в 1. Таким образом, цикломатическое число графа равно $18 - 13 + 2 * 1 = 7$. Значит необходимо рассмотреть 7 линейно независимых циклов и путей.

- 4-7 (1 ветвление)
- 5-8 (1 ветвление)
- 11-12 (1 ветвление)
- 1-4-7-10-11-12-13 (3 ветвления)
- 1-2-3-5-8-10-11-12-13 (5 ветвлений)
- 1-2-3-5-6-9-11-12-13 (4 ветвления)
- 1-2-6-9-11-12-13 (3 ветвления)

Итого сложность равна: $1 + 1 + 1 + 3 + 5 + 4 + 3 = 18$.

Программный расчет оказался невозможным из-за внутренней ошибки программы – программа не может обработать данную структуру графа. Скорее

всего это происходит из-за замыкания цикла на участке 8-5 на вершине 5, которая сама является ветвлением. Ошибки представлены на рис. 2, 3 и 4. Структура графа представлена в приложении А.

```
Bad Graph structure at or near node 5
Please, check the description.
Press any key to continue.
```

Рисунок 2 – Ошибка запуске программы

```
Min ways.... Path #1 -----
->
-----Press a key to continue -----
```

Рисунок 2 – Ошибка при минимальном покрытие графа задания

Рисунок 4 – Ошибка при покрытии графа задания с помощью цикломатического цикла

Далее проведем аналогичные расчеты для программы на языке Си из первой лабораторной работы. Для этого сначала необходимо составить граф. Он представлен на рис. 5.

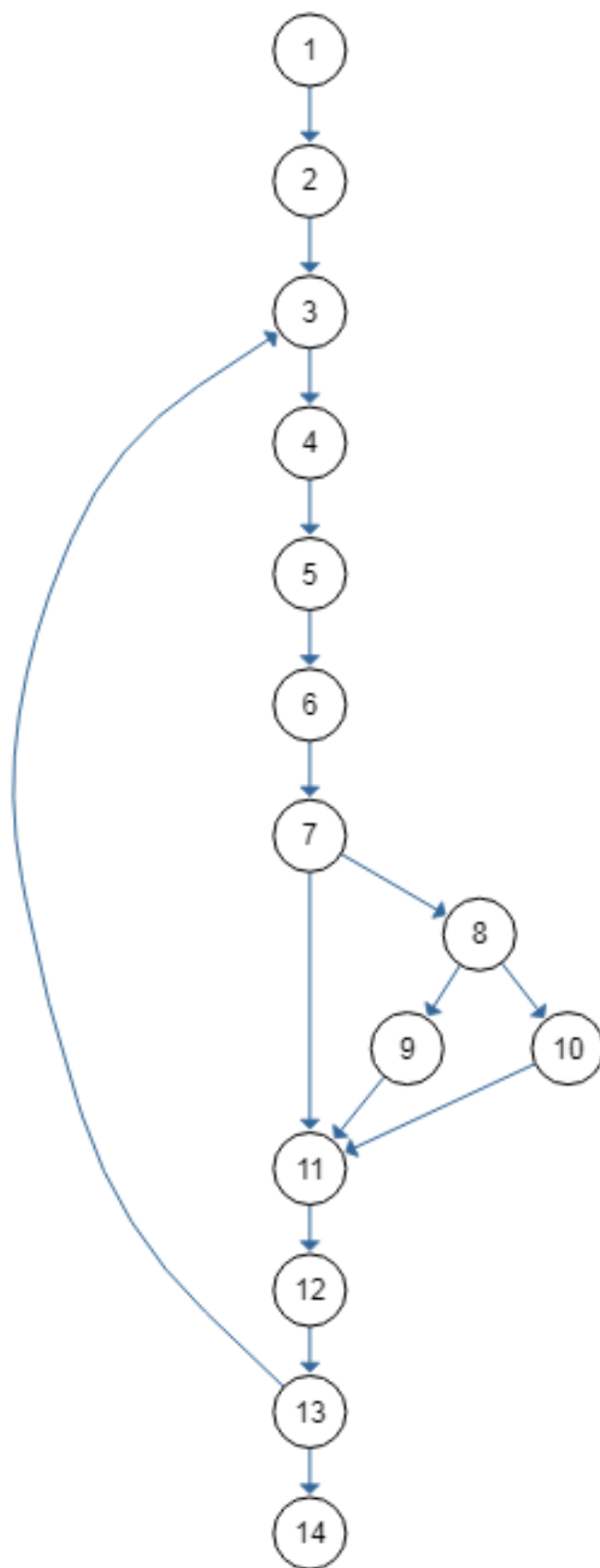


Рисунок 5 – Граф программы

Ключевые узлы графа:

- 2 – передача управления в newton
- 4 – передача управления в func
- 3-13 – цикл do ... while ...
- 7 – проверка на $dfx < tol$
- 8 – проверка на $dfx > 0$

Произведем ручной расчет с помощью минимального покрытия дуг графа.

Ветвления – 7, 8, 13. Всего – 3.

Пути:

- 1-2-3-4-5-6-7-11-12-13-3-4-5-6-7-8-9-11-12-13-3-4-5-6-7-8-10-11-12-13-14 (8 ветвлений)

Итого сложность равна: 8.

Произведем программный расчет. Результаты на рис. 6.

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

Рисунок 6 – Минимальное покрытие программного графа

Произведем ручной расчет сложности на основе цикломатического числа. Число вершин – 14, число ребер – 16, для того чтобы сделать граф связным, необходимо добавить одно ребро из 14 в 1. Таким образом, цикломатическое число графа равно $16 - 14 + 2 * 1 = 4$.

Пути:

- 3-4-5-6-7-11-12-13 (2 ветвления)
- 1-2-3-4-5-6-7-11-12-13-14 (2 ветвления)
- 1-2-3-4-5-6-7-8-9-11-12-13-14 (3 ветвления)
- 1-2-3-4-5-6-7-8-10-11-12-13-14 (3 ветвления)

Итого сложность равна: $2 + 2 + 3 + 3 = 10$.

Произведем программный расчет Результаты на рис.7.

```

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

```

Рисунок 7 – Покрытие программного графа на основе цикломатического цикла

Мы можем убедиться, что в обоих случаях измерения совпадают с точностью до порядка путей. Структура графа для программы представлена в приложении Б.

Выводы.

В ходе выполнения данной лабораторной работы были изучены критерии оценивания структурной сложности программ. Была проведена ручная и автоматическая оценка структурной сложности программ. Кроме того, на примере программы из предыдущей лабораторной были изучены и применены на практике принципы построения структурного графа программы.

ПРИЛОЖЕНИЕ А

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

ПРИЛОЖЕНИЕ Б

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