

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»

Студентка гр. 6304

Иванкова В.М.

Преподаватель

Кирияничиков 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.

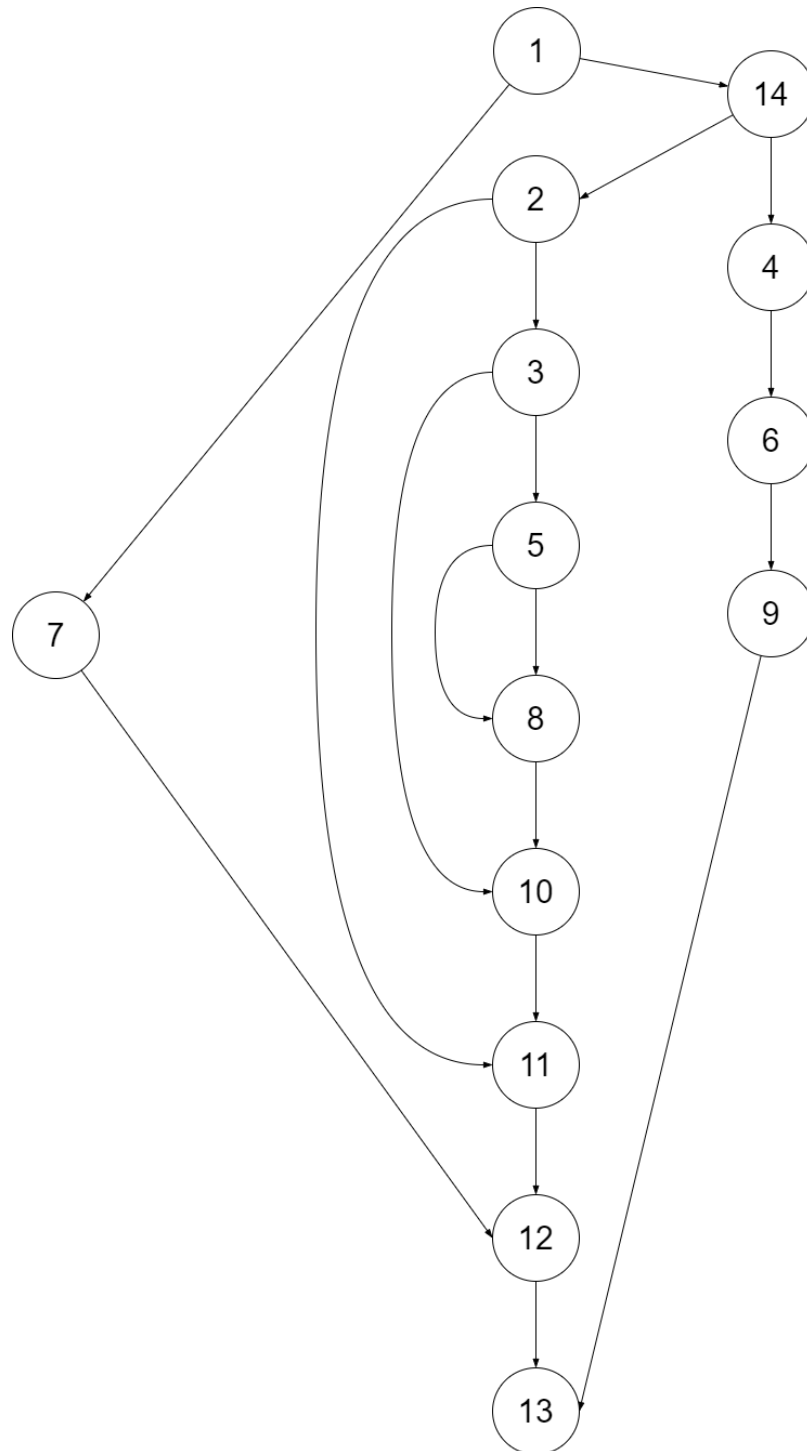
Варианты программ:

- 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 6.



Ход работы

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

1.1. Вручную

Ветвления: 1, 2, 3, 5, 14.

Минимальный набор путей:

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

Сложность равна 14

1.2. С помощью программы ways.exe

Граф для программы:

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

```
Top{1}
```

```
Last{13}
```

```
Arcs{
```

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

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

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

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

```
arc(3,5);
```

```
arc(3,8);
```

```
arc(3,11);
```

```
arc(4,6);
```

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

```
arc(5,7);
```

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

```
arc(7,9);
```

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

```
arc(8,11);
```

```
arc(9,10);
```

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

```
arc(11,12);
```

```
arc(12,13);
```

```
}
```

Результат работы программы:

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

Complexity = 14
Press a key...
```

1.3. Сравнение результатов

Все пути и сложность совпадают.

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

2.1. Вручную

Количество рёбер – 18.

Количество вершин – 14.

Цикломатическое число равно $= 18 - 14 + 2 \cdot 1 = 6$.

Ветвления: 1, 2, 3, 5, 14.

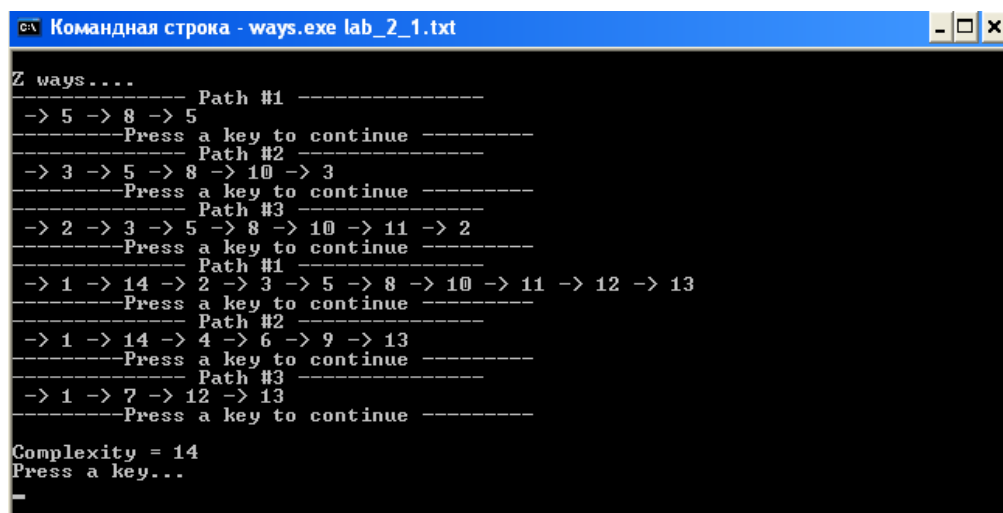
Набор путей:

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

Сложность равна 14.

2.2. С помощью программы

Результат работы программы:



```
Командная строка - ways.exe lab_2_1.txt
Z ways....
----- Path #1 -----
-> 5 -> 8 -> 5
Press a key to continue -----
----- Path #2 -----
-> 3 -> 5 -> 8 -> 10 -> 3
Press a key to continue -----
----- Path #3 -----
-> 2 -> 3 -> 5 -> 8 -> 10 -> 11 -> 2
Press a key to continue -----
----- Path #1 -----
-> 1 -> 14 -> 2 -> 3 -> 5 -> 8 -> 10 -> 11 -> 12 -> 13
Press a key to continue -----
----- Path #2 -----
-> 1 -> 14 -> 4 -> 6 -> 9 -> 13
Press a key to continue -----
----- Path #3 -----
-> 1 -> 7 -> 12 -> 13
Press a key to continue -----
Complexity = 14
Press a key...
_
```

2.3. Сравнение результатов.

Все пути и сложность совпадают.

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

Kod:

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

void swap (float *x, float *y)
{
    float temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

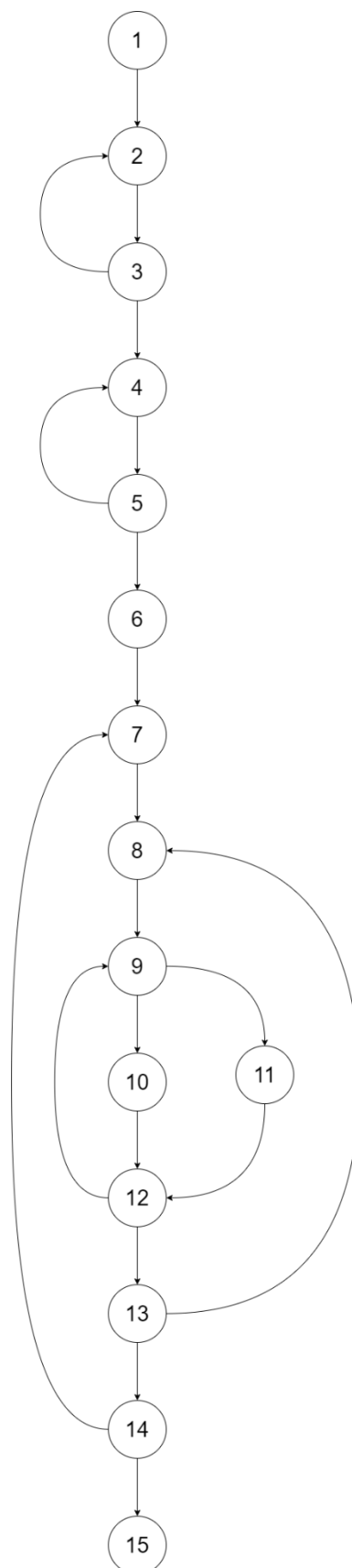
void shellsort(float arr[], int num)
{
    int i, j, k;
    for (i = num / 2; i > 0; i = i / 2)
    {
        for (j = i; j < num; j++)
        {
            for(k = j - i; k >= 0; k = k - i)
            {
                if (arr[k+i] >= arr[k])
                    break;
                else
                {
                    swap(&arr[k], &arr[k+i]);
                }
            }
        }
    }
}

void write_arr(float arr[], int num)
{
    int i;
    for (i = 0; i < num; i++)
    {
        printf("%f ", arr[i]);
    }
}

int main()
{
    int num = 80;
    float my_max = 100.0;
    float arr[num];
    int k;

    for (k = 0 ; k < num; k++)
    {
        arr[k] = (float)rand()/(float)(RAND_MAX/my_max);
    }
    shellsort(arr, num);
    write_arr(arr, num);
    return 0;
}
```

Граф:



3.1. Вручную

Ветвления: 3,5,9,12,13,14.

Минимальный набор путей:

1) 1-2-3-2-3-4-5-4-5-6-7-8-9-10-12-9-11-12-13-8-9-10-12-13-14-7-8-9-10-12-13-14-15 (17 ветвлений)

Сложность равна 17.

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}
```

```
Top{1}
```

```
Last{15}
```

```
Arcs{
```

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

```
arc(2,3);
```

```
arc(3,2);
```

```
arc(3,4);
```

```
arc(4,5);
```

```
arc(5,4);
```

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

```
arc(12,13);
```

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

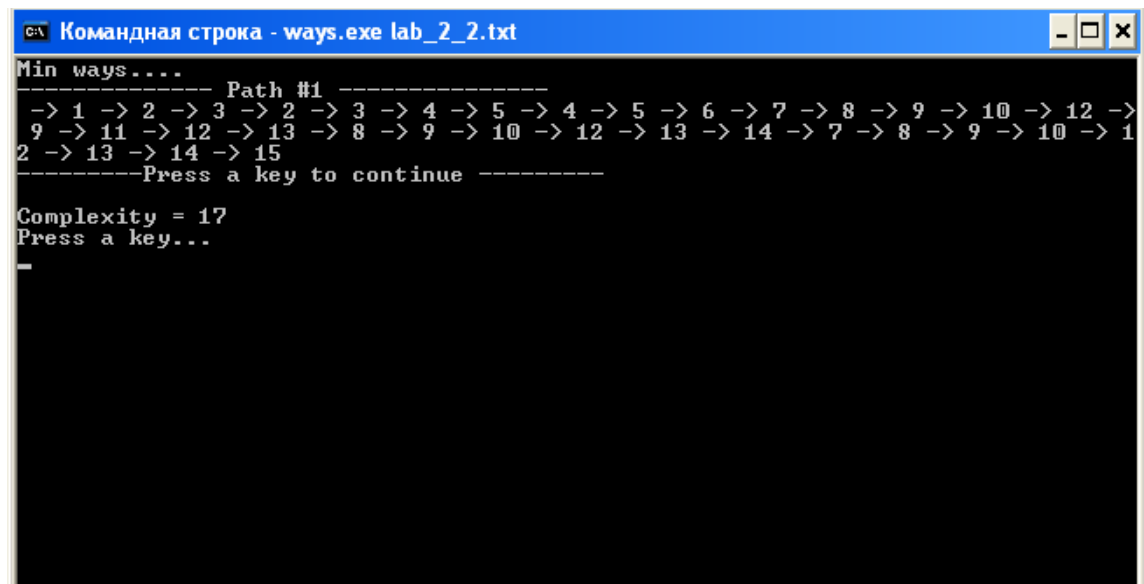
```
arc(13,14);
```

```
arc(14,7);
```

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

```
}
```

Результат работы программы:



```
C:\> Командная строка - ways.exe lab_2_2.txt

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

Complexity = 17
Press a key...
-
```


3.3. Сравнение результатов.

Все пути и сложность совпадают.

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

4.1. Вручную

Количество рёбер – 20.

Количество вершин – 15.

Цикломатическое число равно $= 20 - 15 + 2 \cdot 1 = 7$.

Ветвления: 3, 5, 9, 12, 13, 14.

Набор путей:

- 1) 2-3-2 (1 ветвление)
- 2) 4-5-4 (1 ветвление)
- 3) 9-10-12-9 (3 ветвления)
- 4) 8-9-10-12-13-8 (3 ветвления)
- 5) 7-8-9-10-12-13-14-7 (4 ветвления)
- 6) 1-2-3-4-5-6-7-8-9-10-12-13-14-15 (6 ветвлений)
- 7) 1-2-3-4-5-6-7-8-9-11-12-13-14-15 (6 ветвлений)

Сложность равна 24.

4.2. С помощью программы ways.exe.

Результат работы программы:

```
Командная строка - ways.exe lab_2_2.txt
----- Path #1 -----
-> 2 -> 3 -> 2
-----Press a key to continue -----
----- Path #2 -----
-> 4 -> 5 -> 4
-----Press a key to continue -----
----- Path #3 -----
-> 9 -> 10 -> 12 -> 9
-----Press a key to continue -----
----- Path #4 -----
-> 8 -> 9 -> 10 -> 12 -> 13 -> 8
-----Press a key to continue -----
----- Path #5 -----
-> 7 -> 8 -> 9 -> 10 -> 12 -> 13 -> 14 -> 7
-----Press a key to continue -----
----- Path #1 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 10 -> 12 -> 13 -> 14 -> 15
-----Press a key to continue -----
----- Path #2 -----
-> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> 11 -> 12 -> 13 -> 14 -> 15
-----Press a key to continue -----

Complexity = 24
Press a key...
```

4.3. 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.