Оценка структурной и временной сложности программ

Домашнее задание Сложность-2

Задача 1

1. Исходный текст программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication43

{

class Program

{

static void Main(string[] args)

{

double x, y = 0;// аргумент и значение функции.

char rep;// признак повтора.

string str; //Строка для ввода и вывода данных.

Metka://Начало программы.

Console.Clear();// Очистка экрана

Console.ForegroundColor = ConsoleColor.Yellow;//Задание тексту желтого цвета.

Console.WriteLine("Введите аргумент x");// Ввод и считывание

str = Console.ReadLine();// аргумента.

x = double.Parse(str);

if (x<-1||x>1) y=1;// Задание условия.

if (x>=-1 && x<0) y=0.3;

if (x>=0 && x<=1) y=x\*x;

str="F("+x+")="+y;//Формирование строки.

Console.WriteLine(str);//Вывод строки

Console.Beep(1600, 999);

Console.Write("Для повтора вычисления нажмите клавишу k:");//Повторное вычисление при нажатии клавиши k.

rep=char.Parse(Console.ReadLine());

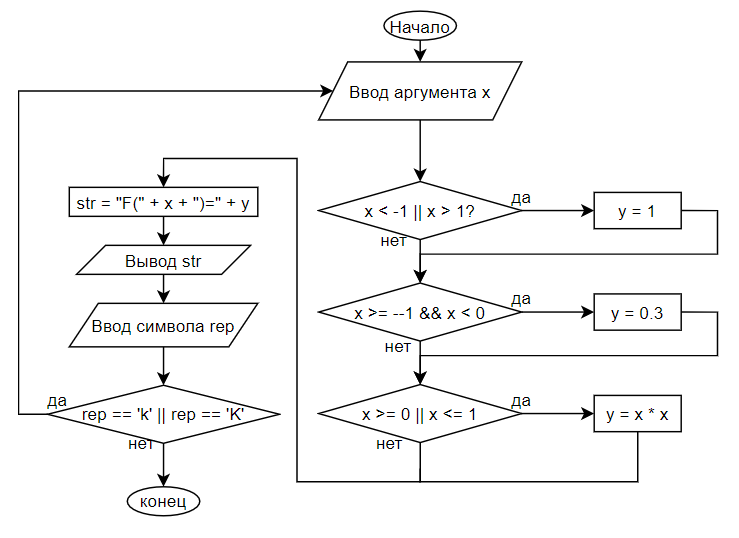
if (rep=='k'||rep=='K') goto Metka;

}

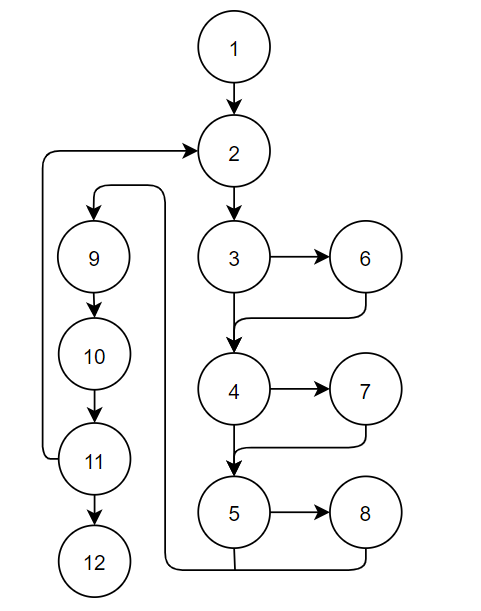
}

}

1. Схема алгоритма

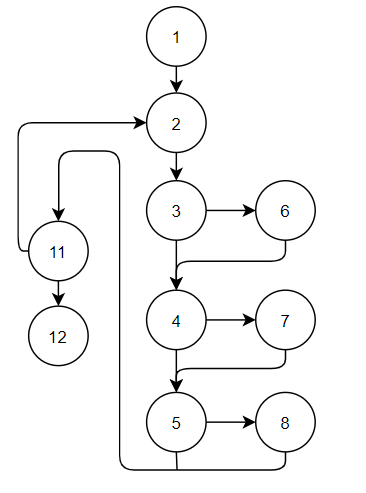


1. Исходный управляющий граф



1. Упрощенный управляющий граф и обоснование его упрощения

=15 - 12 + 2 = 5



=13 - 10 + 2 = 5

Так как цикломатические числа исходного и упрощенного графа равны, можно сказать, что модификация корректна.

1. Оценка алгоритмической сложности

Первый критерий

m1: 1-2-**3**-6-**4**-7-**5**-8-**11**-12; p = 4

m2: 1-2-**3**-**4**-**5**-**11**-2-**3**-**4**-**5**-**11**-12; p = 8

S1 = p1 + p2 = 12

Второй критерий

Ациклические маршруты:

m1 = 1-2-**3**-**4**-**5**-**11**-12; p1 = 4

m2 = 1-2-**3**-6-**4**-**5**-**11**-12; p2 = 4

m3 = 1-2-**3**-**4**-7-**5**-**11**-12; p3 = 4

m4 = 1-2-**3**-6-**4**-7-**5**-**11**-12; p4 = 4

m5 = 1-2-**3**-6-**4**-**5**-8-**11**-12; p5 = 4

m6 = 1-2-**3**-6-**4**-7-**5**-8-**11**-12; p6 = 4

Циклические маршруты:

m7 = 2-**3**-**4**-**5**-**11**; p7 = 4

m8 = 2-**3**-6-**4**-**5**-**11**; p8 = 4

m9 = 2-**3**-**4**-7-**5**-**11**; p9 = 4

m10 = 2-**3**-6-**4**-7-**5**-**11**; p10 = 4

m11 = 2-**3**-6-**4**-**5**-8-**11**; p11 = 4

m12 = 2-**3**-6-**4**-7-**5**-8-**11**; p12 = 4

S2 = 48

Третий критерий:

m1 = 1-2-**3**-**4**-**5**-**11**-12; p1 = 4

m2 = 1-2-**3**-6-**4**-**5**-**11**-12; p2 = 4

m3 = 1-2-**3**-**4**-7-**5**-**11**-12; p3 = 4

m4 = 1-2-**3**-6-**4**-7-**5**-**11**-12; p4 = 4

m5 = 1-2-**3**-6-**4**-**5**-8-**11**-12; p5 = 4

m6 = 1-2-**3**-6-**4**-7-**5**-8-**11**-12; p6 = 4

m7 = 1-2-**3**-**4**-**5**-**11**-2-**3**-**4**-**5**-**11**-12; p7 = 8

m8 = 1-2-**3**-6-**4**-**5**-**11**-2-**3**-6-**4**-**5**-**11**-12; p8 = 8

m9 = 1-2-**3**-**4**-7-**5**-**11**-2-**3**-**4**-7-**5**-**11**-12; p9 = 8

m10 = 1-2-**3**-6-**4**-7-**5**-**11**-2-**3**-6-**4**-7-**5**-**11**-12; p10 = 8

m10 = 1-2-**3**-6-**4**-7-**5**-**11**-2-**3**-6-**4**-7-**5**-**11**-12; p10 = 8

m11 = 1-2-**3**-6-**4**-**5**-8-**11**-**3**-6-**4**-**5**-8-**11**-12; p11 = 8

m12 = 1-2-**3**-6-**4**-7-**5**-8-**11**-2-**3**-6-**4**-7-**5**-8-**11**-12; p12 = 8

S3 = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11 + p12 = 72

1. Матрица смежности:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 11 | 12 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  |  |  |  |  | 1 |  |
| 3 |  | 1 |  |  |  |  |  |  |  |  |
| 4 |  |  | 1 |  |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  |  |  |  |
| 6 |  |  | 1 |  | 1 |  |  |  |  |  |
| 7 |  |  |  | 1 |  |  |  |  |  |  |
| 8 |  |  |  |  | 1 |  |  |  |  |  |
| 11 |  |  |  |  | 1 |  |  | 1 |  |  |
| 12 |  |  |  |  |  |  |  |  | 1 |  |

1. Матрица достижимости

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 11 | 12 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 12 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |

Вывод

Исходя из полученных результатов расчета метрик структурной сложности по первому(S1 = 12), второму(S2 = 48) и третьему(S3 = 72) критериям выделения маршрутов, можно сделать вывод, что программа, характеризуемая заданным графом управления, имеет невысокую алгоритмическую сложность, так как количество используемых в тексте операторов условий 4 в присутствии одного цикла, для проверки которых необходимо проверить от 12 до 72 тестовых вариантов исходных данных.

Задача 2

1. Исходный текст программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

double a,

b,

c,

d,

min,

max,

raz;

char str;

do

{

min = 0;

max = 0;

Console.Clear();

Console.WriteLine("Введите значение переменной a");

a = double.Parse(Console.ReadLine ());

Console.WriteLine("Введите значение переменной b");

b = double.Parse(Console.ReadLine ());

Console.WriteLine("Введите значение переменной c");

c = double.Parse (Console.ReadLine ());

Console.WriteLine("Введите значение переменной d");

d = double.Parse(Console.ReadLine ());

min = a;

if (b < min) min = b;

if (c < min) min = c;

if (d < min) min = c;

max = a;

if (b > max) max = b;

if (c > max) max = c;

if (d > max) max = d;

raz = max- min;

Console.WriteLine("Минимальное число" +min);

Console.WriteLine("Максимальное число" +max);

Console.WriteLine("Разность максимального и минимального значения" +raz);

Console.WriteLine("Для повторного вычисления нажмите Y || y");

str = char.Parse(Console.ReadLine());

}

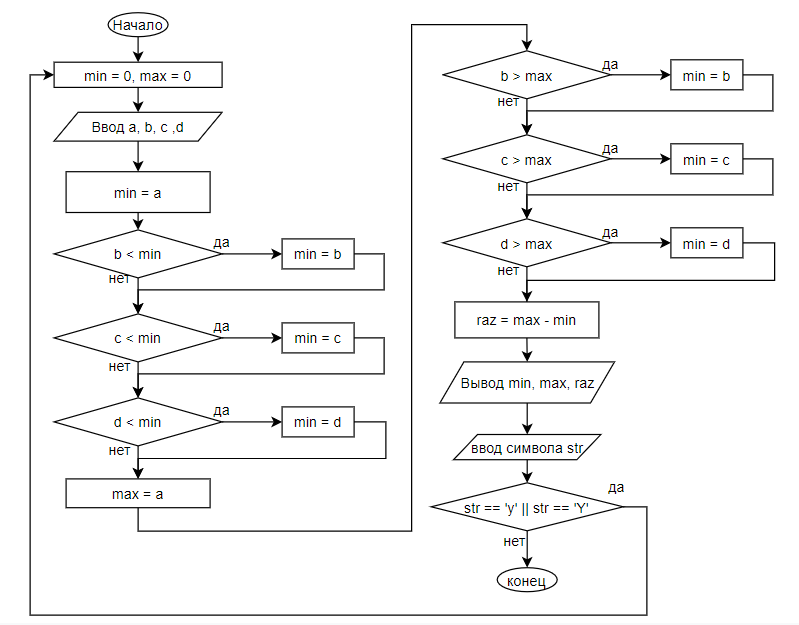
while (str == 'Y'|| str == 'y');

}

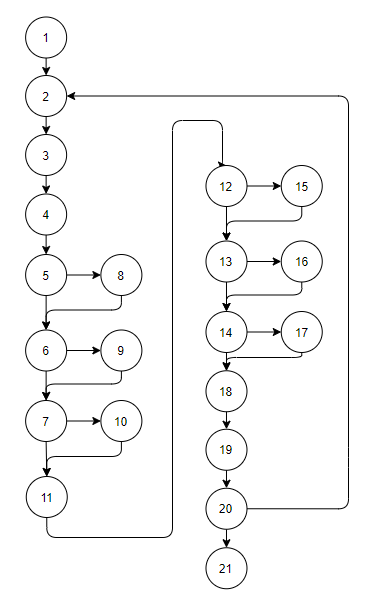
}

}

1. Схема алгоритма

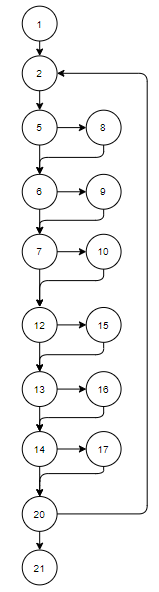


1. Исходный управляющий граф



1. Упрощенный управляющий граф и обоснование его упрощения

= 27 - 21 + 2 = 8



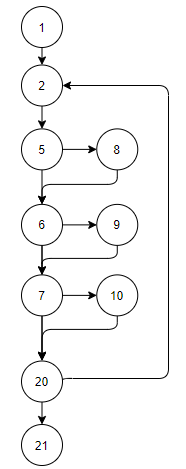
= 22 - 16 + 2 = 8

Так как цикломатические числа исходного и упрощенного графа равны, можно сказать, что модификация корректна.

1. Оценка алгоритмической сложности

Если обратить внимание на исходный текст программы и схему алгоритма, то видно, что операции вычисления минимума и максимума взаимно независимы – результат выполнения одной операции не участвует в выполнении другой (каждая имеет свою сложность), и в ходе рефакторинга можно было бы выделить две таких функции. Каждая из этих функций имела бы единственную ответственность и подвергалась бы тестированию независимо от других участков кода. Видно, что структуры управляющего графа двух этих операций одинаковы (вершины 5-10 и 12-17) и в качестве итоговой сложности можно взять двойную сложность поиска минимума/максимума, так как эти операции выполняются последовательно и независимо друг от друга. Таким образом, далее определяется алгоритмическая сложность минимума/максимума трех аргументов.

Управляющий граф алгоритма определения минимума/максимума трех аргументов:



Видно, что данный граф такой же, как и упрощенный управляющий граф первого задания. Тем не менее, чтобы проверить правильность расчета проводится повторное вычисление сложности:

Первый критерий

m1: 1-2-**5**-**6**-**7**-**20**-2-**5**-**6**-**7**-**20**-21; p1 = 8

m2: 1-2-**5**-8-**6**-9-**7**-**20**-21; p2 = 4

S1 = p1 + p2 = 12

Второй критерий

Ациклические маршруты:

m1 = 1-2-**5**-**6**-**7**-**20**-21; p1 = 4

m2 = 1-2-**5-**8-**6**-**7**-**20**-21; p2 = 4

m3 = 1-2-**5**-**6**-9-**7**-**20**-21; p3 = 4

m4 = 1-2-**5**-**6**-**7**-10-**20**-21; p4 = 4

m5 = 1-2-**5-**8**-6**-9-**7**-**20**-21; p3 = 4

m6 = 1-2-**5**-8**-6**-9-**7**-10-**20**-21; p4 = 4

Циклические маршруты:

m7 = 2-**5**-**6**-**7**-**11**; p7 = 4

m2 = 2-**5-**8-**6**-**7**-**20**; p2 = 4

m3 = 2-**5**-**6**-9-**7**-**20**; p3 = 4

m4 = 2-**5**-**6**-**7**-10-**20**; p4 = 4

m5 = 2-**5-**8**-6**-9-**7**-**20**; p3 = 4

m6 = 2-**5**-8**-6**-9-**7**-10-**20**; p4 = 4

S2 = 48

Третий критерий:

m1 = 1-2-**3**-**4**-**5**-**11**-12; p1 = 4

m2 = 1-2-**3**-6-**4**-**5**-**11**-12; p2 = 4

m3 = 1-2-**3**-**4**-7-**5**-**11**-12; p3 = 4

m4 = 1-2-**3**-6-**4**-7-**5**-**11**-12; p4 = 4

m5 = 1-2-**3**-6-**4**-**5**-8-**11**-12; p5 = 4

m6 = 1-2-**3**-6-**4**-7-**5**-8-**11**-12; p6 = 4

m7 = 1-2-**3**-**4**-**5**-**11**-2-**3**-**4**-**5**-**11**-12; p7 = 8

m8 = 1-2-**3**-6-**4**-**5**-**11**-2-**3**-6-**4**-**5**-**11**-12; p8 = 8

m9 = 1-2-**3**-**4**-7-**5**-**11**-2-**3**-**4**-7-**5**-**11**-12; p9 = 8

m10 = 1-2-**3**-6-**4**-7-**5**-**11**-2-**3**-6-**4**-7-**5**-**11**-12; p10 = 8

m10 = 1-2-**3**-6-**4**-7-**5**-**11**-2-**3**-6-**4**-7-**5**-**11**-12; p10 = 8

m11 = 1-2-**3**-6-**4**-**5**-8-**11**-**3**-6-**4**-**5**-8-**11**-12; p11 = 8

m12 = 1-2-**3**-6-**4**-7-**5**-8-**11**-2-**3**-6-**4**-7-**5**-8-**11**-12; p12 = 8

S3 = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11 + p12 = 72

Результаты сошлись с параметрами первого задания.

1. Матрица смежности:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 21 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  |  |  |  |  | 1 |  |
| 5 |  | 1 |  |  |  |  |  |  |  |  |
| 6 |  |  | 1 |  |  |  |  |  |  |  |
| 7 |  |  |  | 1 |  |  |  |  |  |  |
| 8 |  |  | 1 |  | 1 |  |  |  |  |  |
| 9 |  |  |  | 1 |  |  |  |  |  |  |
| 10 |  |  |  |  | 1 |  |  |  |  |  |
| 20 |  |  |  |  | 1 |  |  | 1 |  |  |
| 21 |  |  |  |  |  |  |  |  | 1 |  |

1. Матрица достижимости

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 21 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 21 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |

Вывод

Исходя из полученных результатов расчета метрик структурной сложности (с учетом того, что мы имеются два алгоритма одинаковой структуры, составляющие исходный алгоритм – их сложности удваиваются) по первому (S1 = 24), второму (S2 = 96) и третьему (S3 = 144) критериям выделения маршрутов, можно сделать вывод, что программа, характеризуемая заданным графом управления, имеет невысокую алгоритмическую сложность, так как количество используемых в тексте операторов условий 7 в присутствии одного цикла, охватывающего все условия, для проверки которых необходимо проверить от 24 до 144 тестовых вариантов исходных данных.

Задача 3

1. Исходный текст программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Collections;

namespace ConsoleApplication1

{

class Method

{

public double[,] Заполнение(int n, int m)

{

double[,] a = new double[n,m];

int i,j;

Random gen = new Random();

for (i = 0; i < n; i++)

for (j = 0; j < m; j++)

a[i, j] = gen.NextDouble() \* (21.7 - (-12.4)) + (-12.4);

return a;

}

public void Вывод(double[,] a, int n, int m)

{

int i,j;

for (i = 0; i < n; i++, Console.WriteLine())

for (j = 0; j < m; j++)

Console.Write("{0,7:f1} ", a[i, j]);

}

public ArrayList Копирование(double[,] a, int n, int m)

{

int i, j;

ArrayList b = new ArrayList();

for (i = 0; i < n; i++)

for (j = 0; j < m; j++)

if (a[i, j] < 0)

b.Add(a[i, j]);

return b;

}

public void Вывод1(ArrayList b)

{

int i;

for (i = 0; i < b.Count; i++)

Console.Write("{0,2:f1} ", b[i]);

}

}

class Program

{

static void Main(string[] args)

{

int n,m;

Console.WriteLine("Введите кол-во строк в массиве: ");

n= int.Parse(Console.ReadLine());

Console.WriteLine("Введите кол-во столбцов в массиве: ");

m= int.Parse(Console.ReadLine());

double[,] a = new double[n, m];

Method obj1 = new Method();

Method obj2 = new Method();

Method obj3 = new Method();

Method obj4 = new Method();

obj1.Вывод(obj2.Заполнение(n, m), n, m);

obj4.Вывод1( obj3.Копирование(obj2.Заполнение(n, m), n, m));

Console.ReadLine();

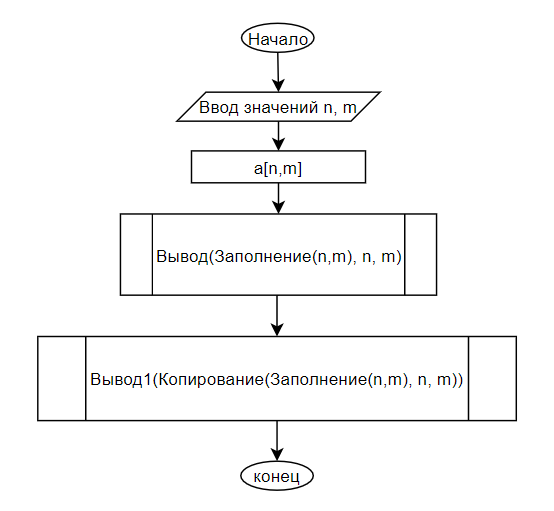
}

}

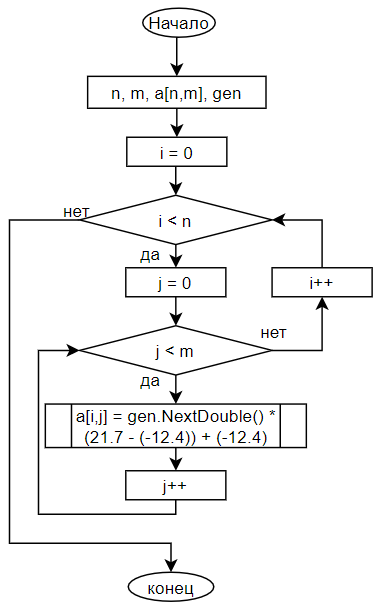
}

1. Схема алгоритма

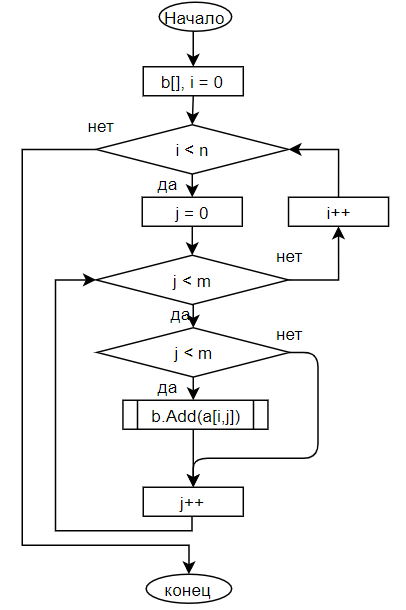
Метод Main:



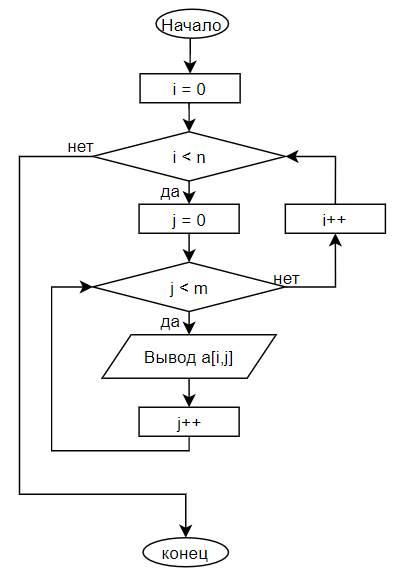
Метод Заполнение(n, m):



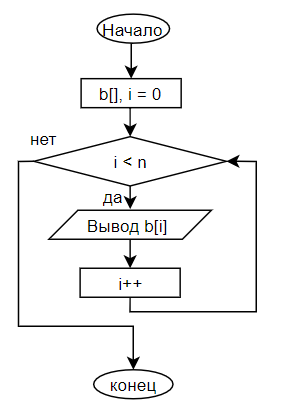
Метод Копирование([,]a, n, m):



Метод Вывод([,]a, n, m):

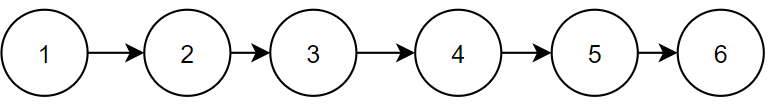


Метод Вывод1(b[])

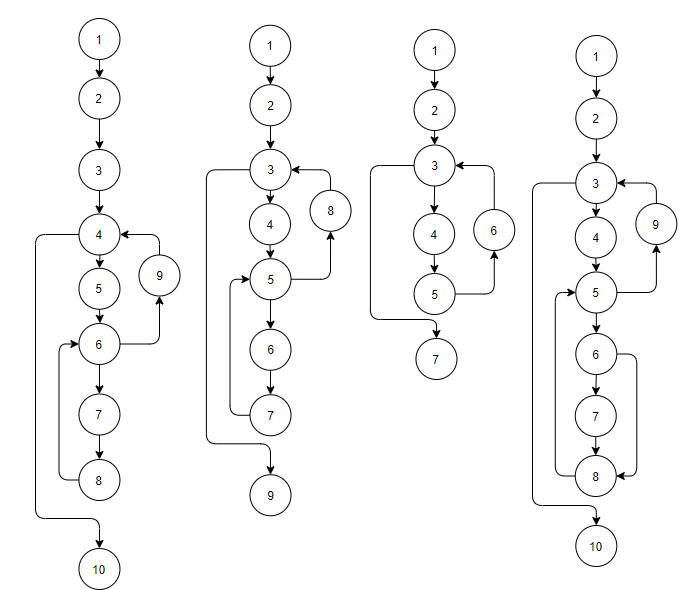


1. Исходный управляющий граф

Метод Main:

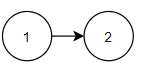


Методы Заполнение, Вывод, Вывод1 и Копирование соответственно:

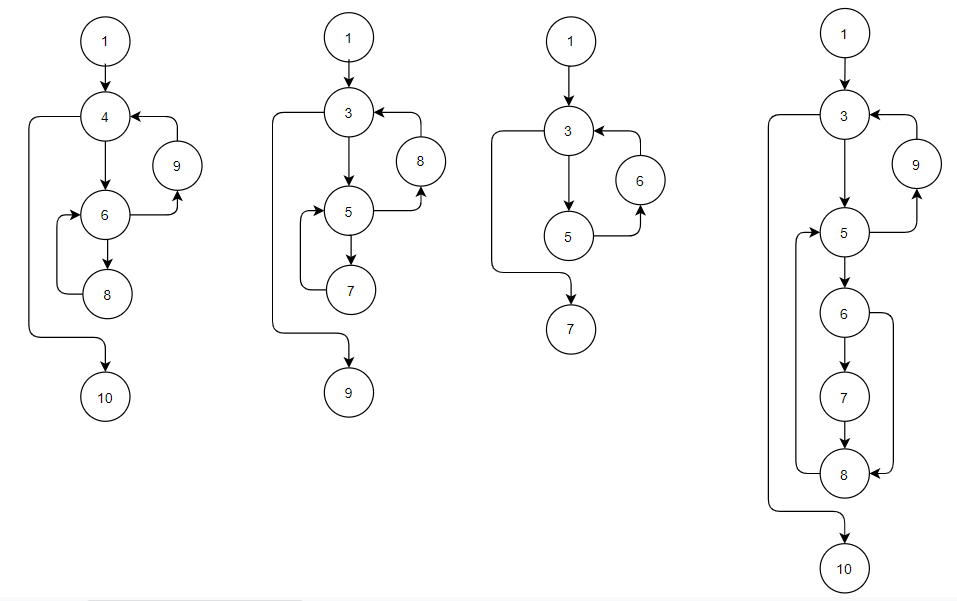


1. Упрощенный управляющий граф и обоснование его упрощения

Метод Main:



Методы Заполнение, Вывод, Вывод1 и Копирование соответственно:



Обоснование Main:

= 5 - 6 + 2 = 1

= 1 - 2 + 2 = 1

Обоснование Заполнение:

= 10 - 9 + 2 = 3

= 7 - 6 + 2 = 3

Обоснование Вывод:

= 10 - 9 + 2 = 3

= 7 - 6 + 2 = 3

Обоснование Вывод1:

= 7 - 7 + 2 = 2

= 5 - 5 + 2 = 2

Обоснование Копирование:

= 11 - 9 + 2 = 4

= 10 - 8 + 2 = 4

Так как цикломатические числа исходных и упрощенных графов равны, можно сказать, что модификации корректны.

1. Оценка алгоритмической сложности

Методы Заполнение и Вывод(одинаковый граф управления без учета нумерации вершин):

Первый критерий

m1: 1-**4**-**6**-8-**6**-9-**4**-10; p1 = 4

S1 = p1= 4

Второй критерий

Ациклические маршруты:

m1 = 1-**4**-10; p1 = 1

m2 = 1-**4**-**6**-8-**6**-9-**4**-10; p2 = 4

m3 = 1-**4**-**6**-9-**4**-10; p3 = 3

Циклические маршруты:

m4 = **4**-**6**-9; p4 = 2

m5 = **6**-8; p5 = 1

S2 = p1 + p2+ p3 + p4 + p5 = 11

Третий критерий:

m1 = 1-**4**-10; p1 = 1

m2 = 1-**4**-**6**-8-**6**-9-**4**-10; p2 = 4

m3 = 1-**4**-**6**-9-**4**-10; p3 = 3

m4 = 1-**4**-**6**-8-**6**-8-**6-**9-**4**-10; p4 = 5

m5 = 1-**4**-**6**-9-**4**-**6**-9-**4**-10; p5 = 5

m6 = 1-**4**-**6**-8-**6**-8-**6**-9-**4**-**6**-9-**4-**10; p6 = 7

S3 = p1 + p2 + p3 + p4 + p5 + p6 = 25

Метод Вывод1

Первый критерий

m1= 1-**3**-5-6-**3**-7; p1 = 2

S1 = p1= 2

Второй критерий

Ациклические маршруты:

m1 = 1-**3**-10; p1 = 1

m2 = 1-**3**-5-6-**3**-7; p2 = 2

Циклические маршруты:

m3 = **3**-5-6; p3 = 2

S2 = p1 + p2+ p3 = 5

Третий критерий:

m1 = 1-**3**-10; p1 = 1

m2 = 1-**3**-5-6-**3**-7; p2 = 2

m3 = 1-**3**-5-6-**3**-5-6-**3**-7; p2 = 3

S3 = p1 + p2 + p3 = 6

Метод Копирование

Первый критерий

m1: 1-**3**-**5**-**6**-7-8-**5**-9-**3**-10; p1 = 5

m2: 1-**3**-**5**-**6**-8-**5**-9-**3**-10; p2 = 5

S1 = p1 + p2 = 10

Второй критерий

Ациклические маршруты:

m1: 1-**3**-**5**-**6**-7-8-**5**-9-**3**-10; p1 = 5

m2: 1-**3**-**5**-**6**-8-**5**-9-**3**-10; p2 = 5

m3: 1-**3**-10; p3 = 1

m4: 1-**3**-**5**-9-**3**-10; p4 = 3

Циклические маршруты:

m5 = **3**-**5**-9; p5 = 2

m6 = **5**-**6**-7-8; p6 = 2

m7 = **5**-**6**-8; p7 = 2

S2 = p1 + p2+ p3 + p4 + p5 + p6 + p7 = 20

Третий критерий:

m1 = 1-**3**-**5**-**6**-7-8-**5**-9-**3**-10; p1 = 5

m2 = 1-**3**-**5**-**6**-8-**5**-9-**3**-10; p2 = 5

m3 = 1-**3**-10; p3 = 1

m4 = 1-**3**-**5**-9-**3**-10; p4 = 3

m5 = 1-**3**-**5**-9-**3**-**5**-9-**3-**10; p5 = 5

m6 = 1-**3**-**5**-**6**-7-8-**5**-**6**-7-8-**5**-9-**3**-10; p6 = 7

m7 = 1-**3**-**5**-**6**-8-**5**-**6**-8-**5**-9-**3**-10; p7 = 7

m8 = 1-**3**-**5**-**6**-8-**5**-**6**-7-8-**5**-9-**3**-10; p8 = 7

m9 = 1-**3**-**5**-**6**-7-8-**5**-**6**-8-**5**-9-**3**-10; p9 = 7

S3 = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 = 47

1. Матрица смежности

Методы Заполнение и Вывод(одинаковый граф управления без учета нумерации вершин):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 4 | 6 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |
| 4 | 1 |  |  |  | 1 |  |
| 6 |  | 1 |  | 1 |  |  |
| 8 |  |  | 1 |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  | 1 |  |  |  |  |

Метод Вывод1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 5 | 6 | 7 |
| 1 |  |  |  |  |  |
| 3 | 1 |  |  | 1 |  |
| 5 |  | 1 |  |  |  |
| 6 |  |  | 1 |  |  |
| 7 |  | 1 |  |  |  |

Метод Копирование

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |
| 3 | 1 |  |  |  |  |  | 1 |  |
| 5 |  | 1 |  |  |  | 1 |  |  |
| 6 |  |  | 1 |  |  |  |  |  |
| 7 |  |  |  | 1 |  |  |  |  |
| 8 |  | 1 |  | 1 | 1 |  |  |  |
| 9 |  |  | 1 |  |  |  |  |  |
| 10 |  | 1 |  |  |  |  |  |  |

1. Матрица достижимости

Методы Заполнение и Вывод (одинаковый граф управления без учета нумерации вершин):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 4 | 6 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 |

Метод Вывод1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 5 | 6 | 7 |
| 1 |  |  |  |  |  |
| 3 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 |

Метод Копирование

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  |  |  |  |  |  |  |  |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Вывод

Были получены результаты расчета метрик структурной сложности программы, составленной из функций Заполнение/Вывод (их сложности одинаковы), Вывод1 и Копирование. Результаты метрик, соответственно, равны: по первому (S1 = 4, 2, 10), второму (S2 = 11, 5, 20) и третьему (S3 = 25, 6, 47) критериям выделения маршрутов. Исходя из расчетов, можно сделать вывод, что программа, характеризуемая заданными графами управления, имеет невысокую алгоритмическую сложность, так как наибольшее количество используемых в тексте операторов условий среди алгоритмов равно 4. Общий диапазон данных, которыми необходимо протестировать 4 алгоритма исходной программы равен от 20 до 113 тестовых вариантов исходных данных.