

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Качество и метрология программного обеспечения»

Тема: «Построение операционной графовой модели программы (ОГМП)
и расчет характеристик эффективности ее выполнения
методом эквивалентных преобразований»

Студент гр. 8304

Щука А. А.

Преподаватель

Кирияничков В. А.

Санкт-Петербург

2022

Цель работы.

Построение операционной графовой модели программы (ОГМП) и расчет характеристик эффективности ее выполнения методом эквивалентных преобразований.

Задание.

Для задания из лабораторных работ 1-3 разработать операционную модель управляющего графа программы на основе схемы алгоритма. При выполнении работы рекомендуется для упрощения обработки графа исключить диалог при выполнении операций ввода-вывода данных, а также привести программу к структурированному виду.

Выбрать вариант графа с нагруженными дугами, каждая из которых должна представлять фрагмент программы, соответствующий линейному участку или ветвлению. При расчете вероятностей ветвлений, зависящих от распределения данных, принять равномерное распределение обрабатываемых данных в ограниченном диапазоне (например, $[0,100]$ - для положительных чисел или $[-100,100]$ - для произвольных чисел). В случае ветвлений, вызванных проверкой выхода из цикла, вероятности рассчитываются исходя априорных сведений о числе повторений цикла. Сложные случаи оценки вероятностей ветвлений согласовать с преподавателем.

В качестве параметров, характеризующих потребление ресурсов, использовать времена выполнения команд соответствующих участков программы. С помощью монитора Sampler выполнить оценку времен выполнения каждого линейного участка в графе программы.

Полученную ОГМП, представить в виде графа с нагруженными дугами, у которого в качестве параметров, характеризующих потребление ресурсов на дуге ij , использовать тройку $\{P_{ij}, M_{ij}, D_{ij}\}$, где:

P_{ij} - вероятность выполнения процесса для дуги ij ;

M_{ij} - мат. ожидание потребления ресурса процессом для дуги ij ;

D_{ij} - дисперсия потребления ресурса процессом для дуги ij .

В качестве потребляемого ресурса в данной работе рассматривается время процессора, а оценками мат. ожиданий времен для дуг исходного графа следует принять времена выполнения операторов (команд), соответствующих этим дугам участков программы. Дисперсиям исходных дуг следует присвоить нулевые значения.

Получить описание полученной ОГМП на входном языке пакета CSA III в виде поглощающей марковской цепи (ПМЦ) – (англ.) AMC (absorbing Markov chain) и/или эргодической марковской цепи (ЭМЦ) - EMC (ergodic Markov chain).

С помощью предоставляемого пакетом CSA III меню действий выполнить расчет среднего времени и дисперсии времени выполнения как для всей программы, так и для ее фрагментов, согласованных с преподавателем. Сравнить полученные результаты с результатами измерений, полученными в работе 3.

Ход работы.

1. Текст программы (исходный).

Исходный код программы представлен в приложении А.

2. Профилирование.

Код программы для профилирования, разделенной на функциональные участки, представлен в приложении Б.

3. Граф управления программы.

Граф управления был построен на основе разбиения программы на функциональные участки, т.к. подсчёт затрат времени в лабораторной работе №3 осуществлялся на каждом из таких участков отдельно. Граф представлен на рисунке 1.

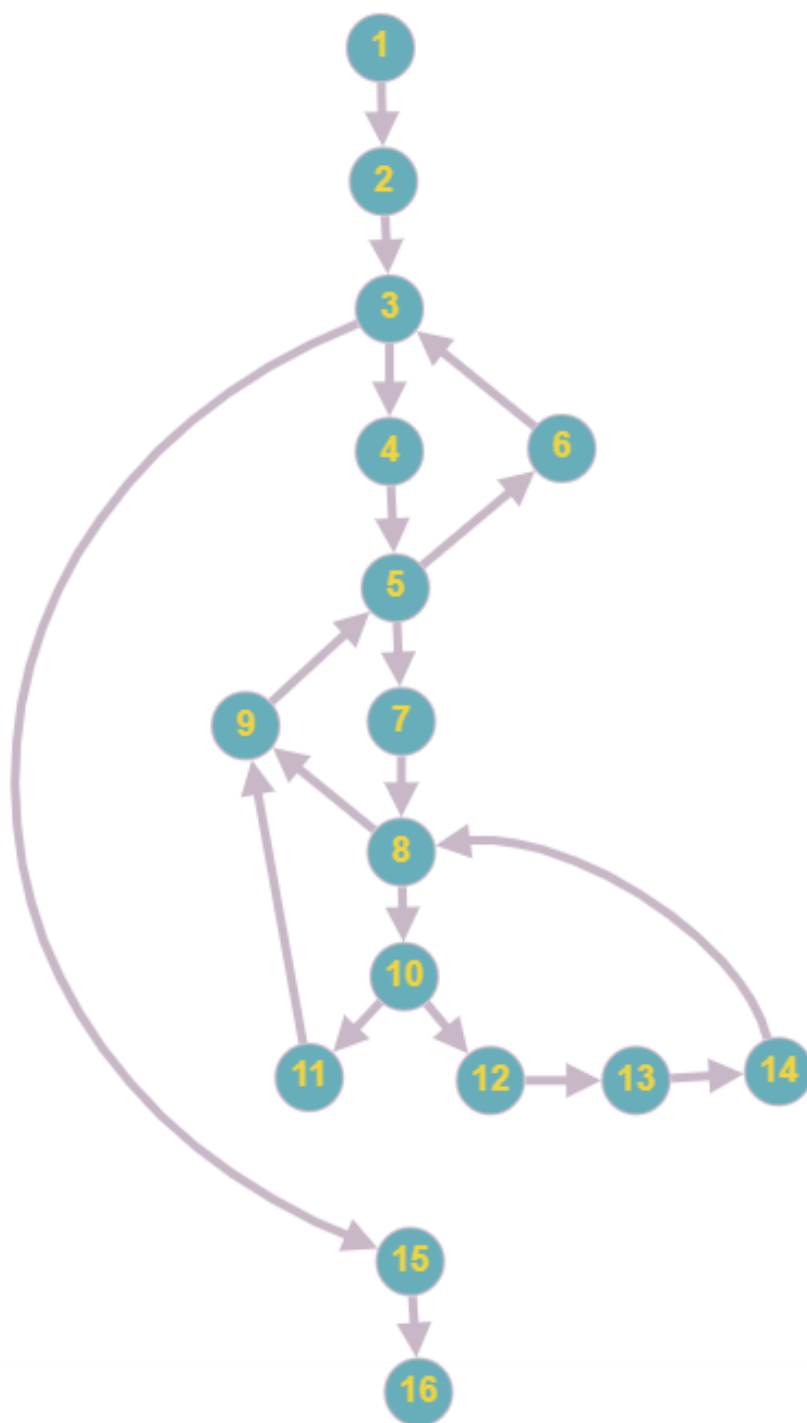


Рисунок 1 - Граф управления

4. Результаты профилирования.

Результаты профилирования из лабораторной работы №3 представлены на рисунке 2.

исх	прием	общее время	кол-во проходов	среднее время
61	15	17.495	1	17.495
15	17	14.684	1	14.684
17	18	48.216	1	48.216
18	20	26.416	7	3.774
18	44	6.742	1	6.742
20	21	48.316	7	6.902
21	23	1526.642	1203	1.269
21	42	73.374	7	10.482
23	24	3538.332	1203	2.941
24	26	4770.774	2232	2.137
24	40	794.921	106	7.499
26	34	9153.063	1135	8.064
26	29	1363.889	1097	1.243
34	36	11585.653	1135	10.208
36	38	2705.705	1135	2.384
38	24	1530.379	1135	1.348
40	21	1148.305	1203	0.955
29	40	2693.816	1097	2.456
42	18	13.505	7	1.929
44	63	59.789	1	59.789

Рисунок 2 – Результаты профилирования

5. Расчет вероятностей и затрат ресурсов для дуг управляющего графа.

Результаты расчета представлены в таблице 1.

Таблица 1 – Расчет вероятностей и затрат ресурсов для дуг.

	Номера строк	Количество проходов	Вероятность
$L_{1-2} = 32,179$	61:15, 15:17	1	1
$L_{2-3} = 48,216$	17:18	1	1
$L_{3-4} = 3,774$	18:20	7	0,875
$L_{3-15} = 6,742$	18:44	1	0,125
$L_{4-5} = 6,902$	20:21	7	1
$L_{5-6} = 10,482$	21:42	7	0,0058
$L_{5-7} = 1,269$	21:23	1203	0.9942
$L_{7-8} = 2,941$	23:24	1203	1
$L_{8-10} = 2,137$	24:26	2232	0,955
$L_{8-9} = 7,499$	24:40	106	0,045
$L_{9-5} = 0,955$	40:21	1203	1
$L_{10-11} = 1,243$	26:29	1097	0,491
$L_{11-9} = 2,456$	29:40	1097	1
$L_{10-12} = 8,064$	26:34	1135	0,509
$L_{12-13} = 10,208$	34:36	1135	1
$L_{13-14} = 2,384$	36:38	1135	1

Таблица 1 (продолжение) – Расчет вероятностей и затрат ресурсов для дуг.

	Номера строк	Количество проходов	Вероятность
$L_{14-8} = 1,348$	38:24	1135	1
$L_{6-3} = 1,929$	42:18	7	1
$L_{15-16} = 59,789$	44:63	1	1

6. Операционная графовая модель программы.

Операционная графовая модель программы представлена на рисунке 3.

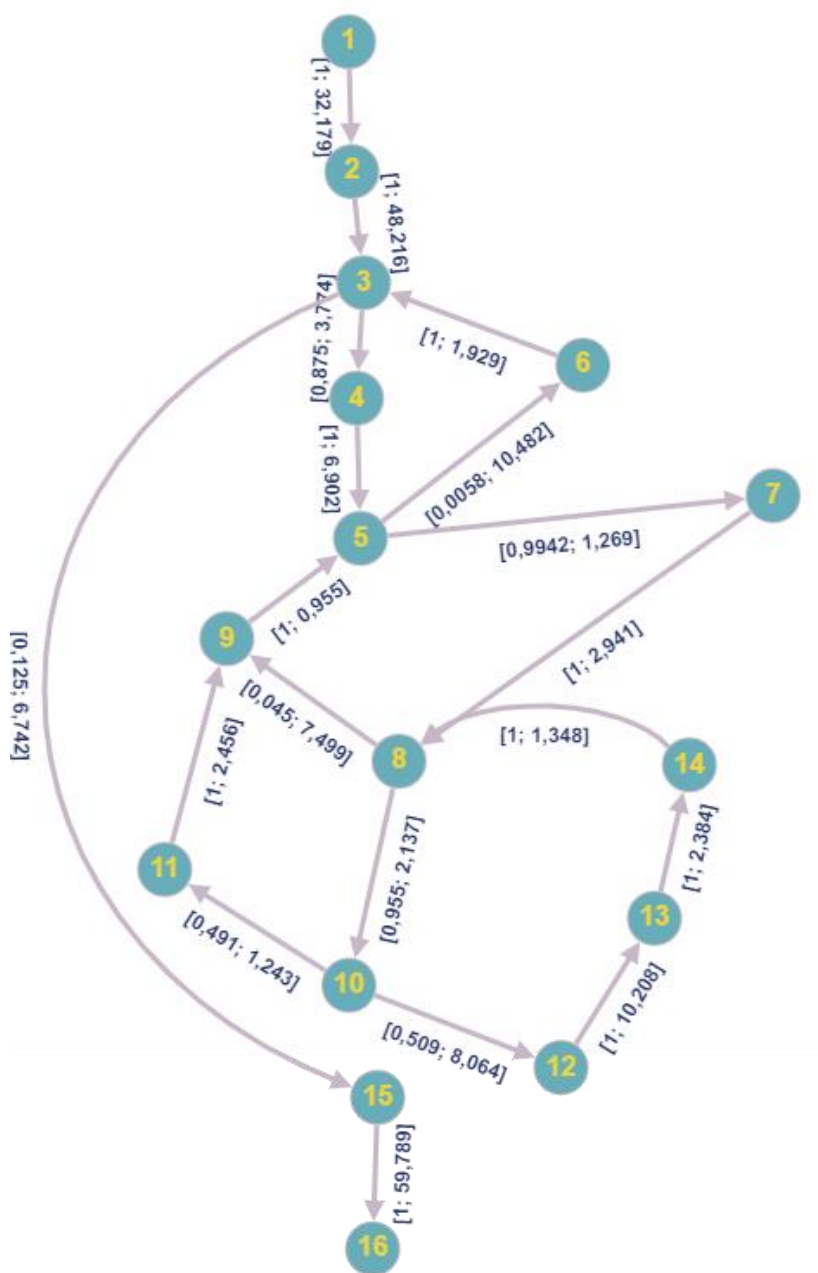


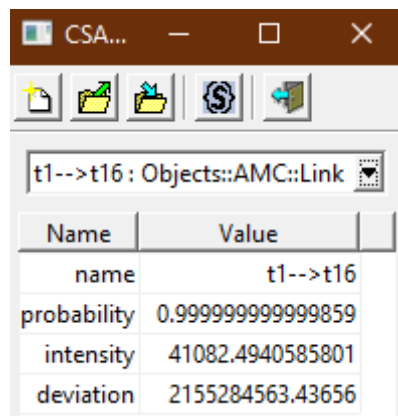
Рисунок 3 - Операционная модель

7. Описание модели model.xml.

Описание модели представлено в приложении В.

8. Результаты.

Результаты работы программы представлены на рисунке 4.



The screenshot shows a window titled 'CSA...' with a toolbar containing icons for file operations and a table of statistics. The table has two columns: 'Name' and 'Value'. The data is as follows:

Name	Value
name	t1-->t16
probability	0.999999999999859
intensity	41082.4940585801
deviation	2155284563.43656

Рисунок 4 - Результат работы программы

Вероятность получилась не равной 1 т.к. существует вычислительная погрешность для чисел с плавающей точкой. Согласно расчётам программы, среднее время выполнения составляет 41082,49 мкс. В пункте 4 данного отчёта приведен результат профилирования программы с использованием SAMPLER_v2, где суммарное время выполнения составило 39971,71 мкс. В итоге, разница между результатами составляет менее 3 %.

Выводы.

В ходе выполнения лабораторной работы была построена операционная графовая модель заданной программы, нагрузочные параметры которой были оценены с помощью профилировщика SAMPLER_v2 и методом эквивалентных преобразований с помощью пакета CSA III были вычислены математическое ожидание и дисперсия времени выполнения как для всей программы, так и для фрагментов программы. Результаты сравнения этих характеристик с полученными в ходе выполнения лабораторной работы №3 согласуются.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <stdio.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]);
                }
            }
        }
    }
}

int main()
{
    const int num = 200;
    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);
    return 0;
}
```


ПРИЛОЖЕНИЕ Б.

КОД ПРОГРАММЫ С РАЗДЕЛЕНИЕМ НА ФУ

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "sampler.h"
4
5  void swap(float* x, float* y)
6  {
7      float temp;
8      temp = *x;
9      *x = *y;
10     *y = temp;
11 }
12
13 void shellsort(float arr[], int num)
14 {
15     SAMPLE;
16     int i, j, k;
17     SAMPLE;
18     for (i = num / 2; SAMPLE, i > 0; i = i / 2)
19     {
20         SAMPLE;
21         for (j = i; SAMPLE, j < num; j++)
22         {
23             SAMPLE;
24             for (k = j - i; SAMPLE, k >= 0; k = k - i)
25             {
26                 SAMPLE;
27                 if (arr[k + i] >= arr[k])
28                 {
29                     SAMPLE;
30                     break;
31                 }
32                 else
33                 {
34                     SAMPLE;
35                     swap(&arr[k], &arr[k + i]);
36                     SAMPLE;
37                 }
38                 SAMPLE;
39             }
40             SAMPLE;
41         }
42         SAMPLE;
43     }
44     SAMPLE;
45 }
46
47
48 int main(int argc, char **argv)
```

```
49  {
50      sampler_init(&argc, argv);
51      const int num = 200;
52      float my_max = 100.0;
53      float arr[num];
54      int k;
55
56      for (k = 0; k < num; k++)
57      {
58          arr[k] = (float)rand() / (float)(RAND_MAX / my_max);
59      }
60
61      SAMPLE;
62      shellsort(arr, num);
63      SAMPLE;
64      return 0;
65  }
```

ПРИЛОЖЕНИЕ В.

ОПИСАНИЕ МОДЕЛИ .XML

```
<model type = "Objects::AMC::Model" name = "model">
  <node type = "Objects::AMC::Top" name = "t1"></node>
  <node type = "Objects::AMC::Top" name = "t2"></node>
  <node type = "Objects::AMC::Top" name = "t3"></node>
  <node type = "Objects::AMC::Top" name = "t4"></node>
  <node type = "Objects::AMC::Top" name = "t5"></node>
  <node type = "Objects::AMC::Top" name = "t6"></node>
  <node type = "Objects::AMC::Top" name = "t7"></node>
  <node type = "Objects::AMC::Top" name = "t8"></node>
  <node type = "Objects::AMC::Top" name = "t9"></node>
  <node type = "Objects::AMC::Top" name = "t10"></node>
  <node type = "Objects::AMC::Top" name = "t11"></node>
  <node type = "Objects::AMC::Top" name = "t12"></node>
  <node type = "Objects::AMC::Top" name = "t13"></node>
  <node type = "Objects::AMC::Top" name = "t14"></node>
  <node type = "Objects::AMC::Top" name = "t15"></node>
  <node type = "Objects::AMC::Top" name = "t16"></node>
  <link type = "Objects::AMC::Link" name = "t1-->t2" probability
= "1.0" intensity = "32.179" deviation = "0.0" source = "t1" dest =
"t2"></link>
  <link type = "Objects::AMC::Link" name = "t2-->t3" probability
= "1.0" intensity = "48.216" deviation = "0.0" source = "t2" dest =
"t3"></link>
  <link type = "Objects::AMC::Link" name = "t3-->t4" probability
= "0.875" intensity = "3.774" deviation = "0.0" source = "t3" dest =
"t4"></link>
  <link type = "Objects::AMC::Link" name = "t4-->t5" probability
= "1.0" intensity = "6.902" deviation = "0.0" source = "t4" dest =
"t5"></link>
  <link type = "Objects::AMC::Link" name = "t5-->t6" probability
= "0.0058" intensity = "10.482" deviation = "0.0" source = "t5" dest =
"t6"></link>
  <link type = "Objects::AMC::Link" name = "t5-->t7" probability
= "0.9942" intensity = "1.269" deviation = "0.0" source = "t5" dest =
"t7"></link>
  <link type = "Objects::AMC::Link" name = "t6-->t3" probability
= "1.0" intensity = "1.929" deviation = "0.0" source = "t6" dest =
"t3"></link>
  <link type = "Objects::AMC::Link" name = "t7-->t8" probability
= "1.0" intensity = "2.941" deviation = "0.0" source = "t7" dest =
"t8"></link>
  <link type = "Objects::AMC::Link" name = "t8-->t9" probability
= "0.045" intensity = "7.499" deviation = "0.0" source = "t8" dest =
"t9"></link>
  <link type = "Objects::AMC::Link" name = "t9-->t5" probability
= "1.0" intensity = "0.955" deviation = "0.0" source = "t9" dest =
"t5"></link>
```

```

    <link type = "Objects::AMC::Link" name = "t8-->t10"
probability = "0.955" intensity = "2.137" deviation = "0.0" source =
"t8" dest = "t10"></link>
    <link type = "Objects::AMC::Link" name = "t10-->t11"
probability = "0.491" intensity = "1.243" deviation = "0.0" source =
"t10" dest = "t11"></link>
    <link type = "Objects::AMC::Link" name = "t11-->t9"
probability = "1.0" intensity = "2.456" deviation = "0.0" source = "t11"
dest = "t9"></link>
    <link type = "Objects::AMC::Link" name = "t10-->t12"
probability = "0.509" intensity = "8.064" deviation = "0.0" source =
"t10" dest = "t12"></link>
    <link type = "Objects::AMC::Link" name = "t12-->t13"
probability = "1.0" intensity = "10.208" deviation = "0.0" source =
"t12" dest = "t13"></link>
    <link type = "Objects::AMC::Link" name = "t13-->t14"
probability = "1.0" intensity = "2.384" deviation = "0.0" source = "t13"
dest = "t14"></link>
    <link type = "Objects::AMC::Link" name = "t14-->t8"
probability = "1.0" intensity = "1.348" deviation = "0.0" source = "t14"
dest = "t8"></link>
    <link type = "Objects::AMC::Link" name = "t3-->t15"
probability = "0.125" intensity = "6.742" deviation = "0.0" source =
"t3" dest = "t15"></link>
    <link type = "Objects::AMC::Link" name = "t15-->t16"
probability = "1.0" intensity = "59.789" deviation = "0.0" source =
"t15" dest = "t16"></link>
</model>

```