

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 расчёт характеристик эффективности её выполнения методом
эквивалентных преобразований

Студент гр. 7304

Есиков О.И.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить возможности построения операционной графовой модели программы (ОГМП) и расчёта характеристик эффективности её выполнения методом эквивалентных преобразований.

Ход выполнения.

Для программы из предыдущих лабораторных работ был построен управляющий граф, представленный на рисунке 1. Исходный код этой программы представлен в Приложении А.

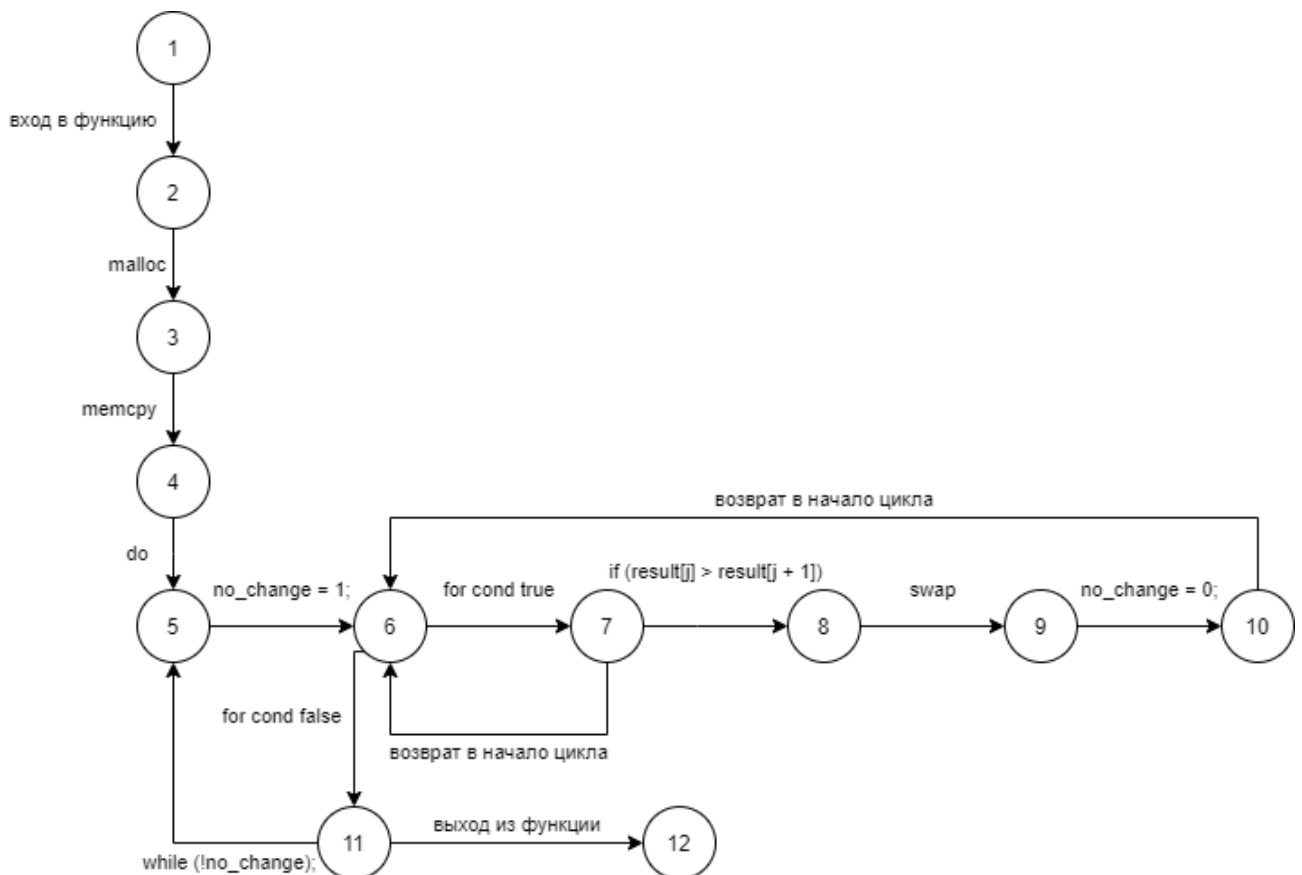


Рисунок 1 – Управляющий граф программы

На основе этой программы была подготовлена программа для профилирования с использованием SAMPLER. Исходный код программы для профилирования представлен в Приложении Б. Результат профилирования представлен на рисунке 2.

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 18	1 : 20	20.95	1	20.95
1 : 20	1 : 22	115.66	1	115.66
1 : 22	1 : 24	0.00	1	0.00
1 : 24	1 : 27	0.00	1	0.00
1 : 27	1 : 29	12.57	68	0.18
1 : 29	1 : 30	39.39	68	0.58
1 : 30	1 : 32	15724.37	5372	2.93
1 : 30	1 : 43	150.86	68	2.22
1 : 32	1 : 41	0.00	3581	0.00
1 : 32	1 : 35	10634.61	1791	5.94
1 : 35	1 : 37	25073.33	1791	14.00
1 : 37	1 : 39	27895.20	1791	15.58
1 : 39	1 : 41	0.00	1791	0.00
1 : 41	1 : 30	0.00	5372	0.00
1 : 43	1 : 27	109.79	67	1.64
1 : 43	1 : 45	1.68	1	1.68
1 : 45	1 : 58	2.51	1	2.51
1 : 56	1 : 18	2.51	1	2.51

Рисунок 2 – Результат профилирования программы

На основании полученных данных с помощью SAMPLER был выполнен расчёт вероятностей и затрат ресурсов для дуг управляющего графа. Результат представлен в таблице 1.

Дуга	Номера строк	Количество проходов	Расчёт вероятности	Затраты ресурсов (среднее время), мкс
L1-L2	56:18	1	1	2.51
L2-L3	18:20	1	1	20.95
L3-L4	20:22	1	1	115.66
L4-L5	24:27	1	1	0.0
L5-L6	27:29	68	1	0.18
L6-L7	30:32	5372	$\frac{5372}{5372+68} = 0.9875$	2.93
L6-L11	30:43	68	$1 - 0.9875 = 0.0125$	2.22
L7-L6	32:41	3581	$\frac{3581}{3581+1791} = 0.6666$	0.0
L7-L8	32:35	1791	$1 - 0.6666 = 0.3334$	5.94
L8-L9	35:37	1791	1	14.0
L9-L10	37:39	1791	1	15.58

L10-L6	41:30	5372	1	0.0
L11-L5	43:27	67	$\frac{67}{67+1} = 0.9853$	1.64
L11-L12	43:35	1	$1 - 0.9853 = 0.0147$	1.68

Таблица 1 – Расчёт вероятностей и затрат ресурсов

На основании произведённых расчётов было выполнено построение операционной графовой модели программы. Результат представлен на рисунке 3.

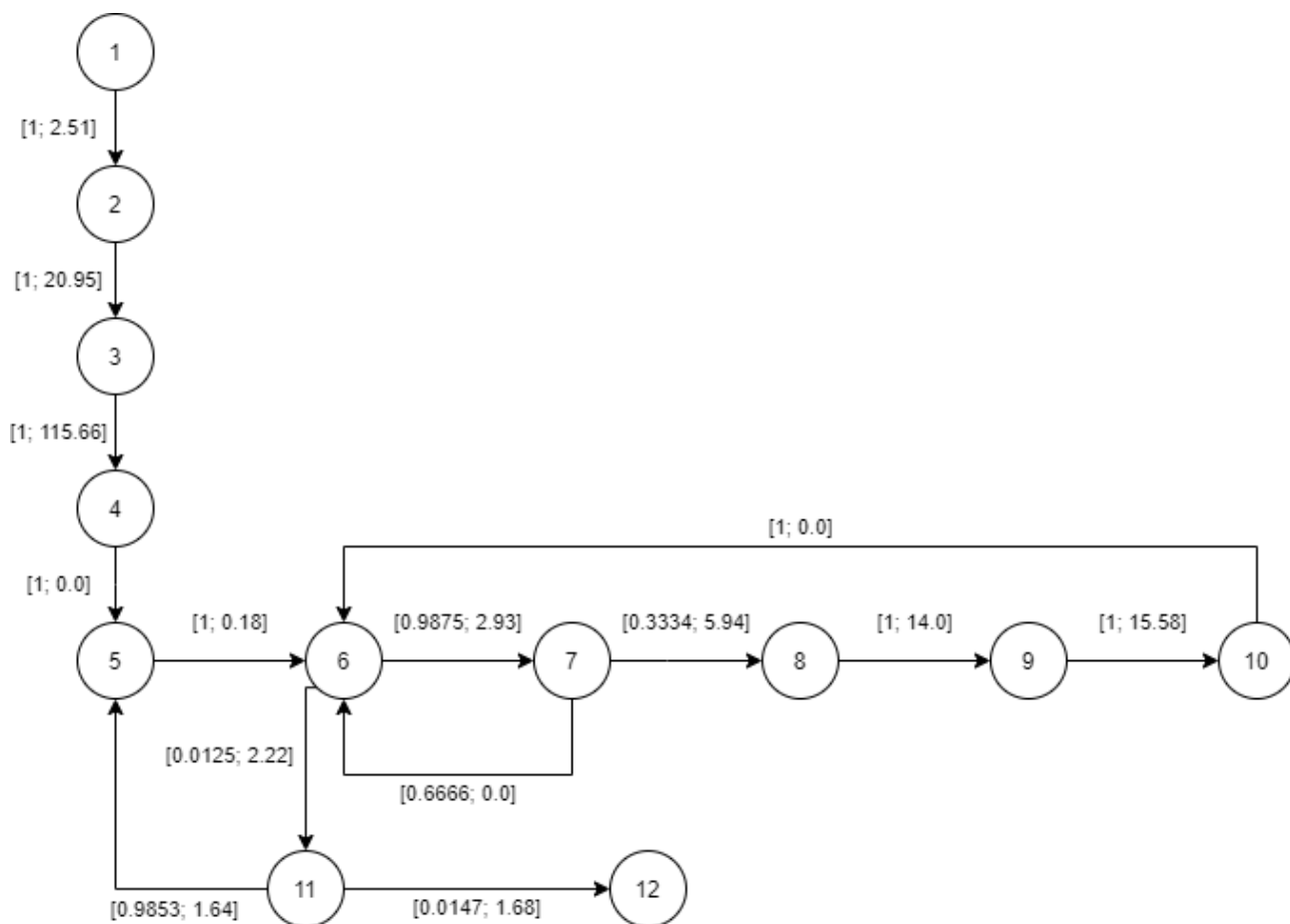


Рисунок 3 – Операционная графовая модель программы

По построенной операционной графовой модели программы был создан XML-файл. Этот файл представлен в Приложении В. Графическое представление модели представлено на рисунке 4.

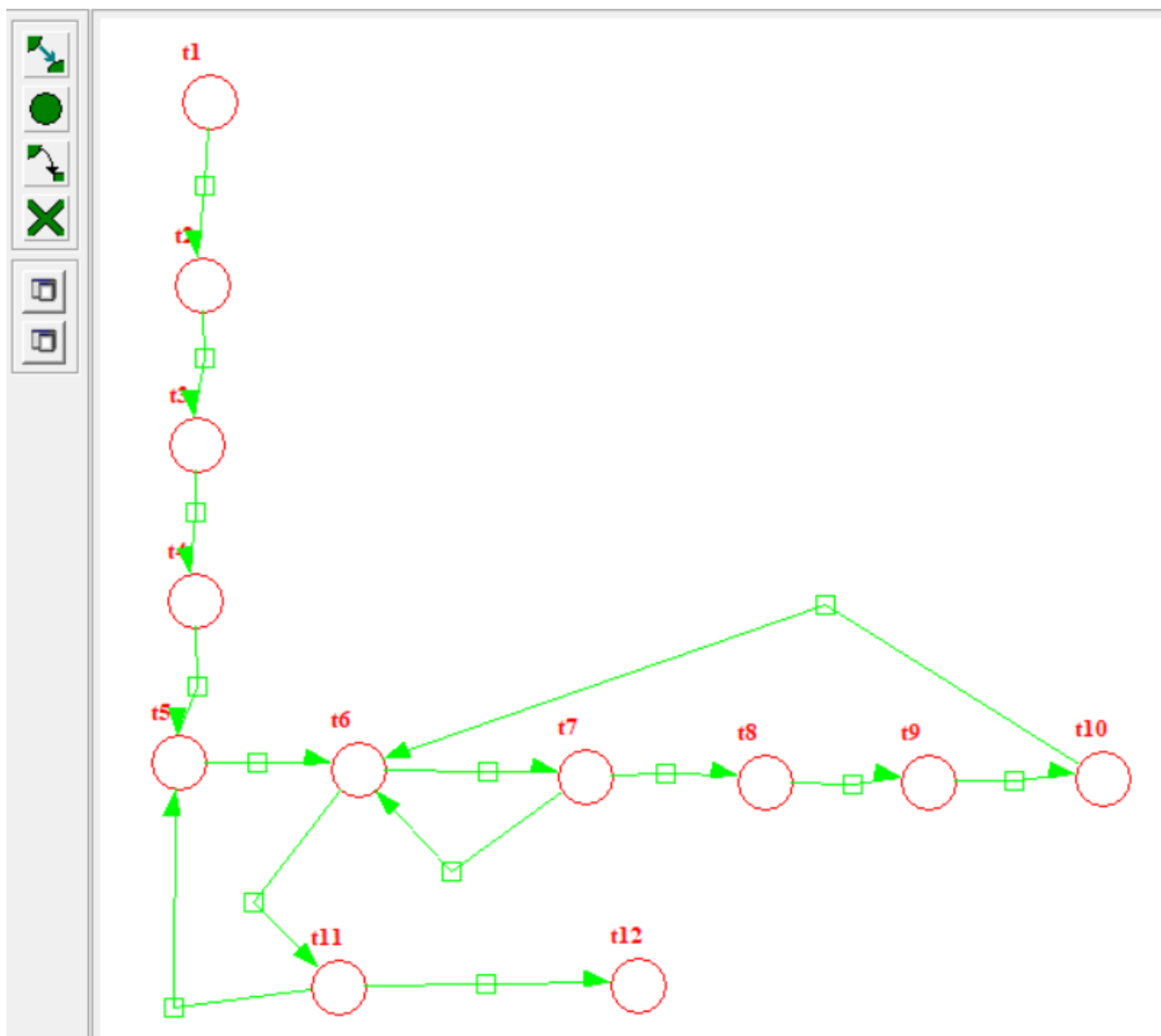


Рисунок 4 – Графическое представление модели

Результат вычисления математического ожидания и дисперсии времени выполнения функции *sort* представлен на рисунке 5.

t1-->t12 : Objects::AMC::Link	
Name	Value
name	t1-->t12
probability	0.9999999999999947
intensity	79802.9063945536
deviation	6348349586.58157

Рисунок 5 – Расчёт с помощью CSAIII

Выводы.

В ходе выполнения лабораторной работы были изучены возможности построения операционной графовой модели программы (ОГМП) и расчёта характеристик эффективности её выполнения методом эквивалентных преобразований. С помощью пакета CSAIII были получены следующие результаты: дисперсия – 6348349586.58 и математическое ожидание – 79802.91, что соответствует результатам, полученным с помощью SAMPLER – 79783.43.

ПРИЛОЖЕНИЯ

Приложение А. Исходный код программы BubbleSort.cpp.

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

#define max 80

void swap(double* p, double* q)
{
    double hold = *p;
    *p = *q;
    *q = hold;
}

double* sort(double* a, int n)
{
    double* result = (double*)malloc(sizeof(double) * n);
    memcpy(result, a, sizeof(double) * n);
    char no_change;
    do
    {
        no_change = 1;
        for (int j = 0; j < n - 1; j++)
        {
            if (result[j] > result[j + 1])
            {
                swap(&result[j], &result[j + 1]);
                no_change = 0;
            }
        }
    } while (!no_change);
    return result;
}

int main()
{
    int n = max;
    double x[max];
    srand(time(NULL));
    for (int i = 0; i < n; i++)
        x[i] = rand() % 100;
    sort(x, n);
    return 0;
}
```

Приложение Б. Исходный код программы BubbleSort_sampler.cpp.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <string.h>
5  #include "Sampler.h"
6
7  #define max 80
8
9  void swap(double* p, double* q)
10 {
11     double hold = *p;
```

```

12     *p = *q;
13     *q = hold;
14 }
15
16 double* sort(double* a, int n)
17 {
18     SAMPLE;
19     double* result = (double*)malloc(sizeof(double) * n);
20     SAMPLE;
21     memcpy(result, a, sizeof(double) * n);
22     SAMPLE;
23     char no_change;
24     SAMPLE;
25     do
26     {
27         SAMPLE;
28         no_change = 1;
29         SAMPLE;
30         for (int j = 0; SAMPLE, j < n - 1; j++)
31         {
32             SAMPLE;
33             if (result[j] > result[j + 1])
34             {
35                 SAMPLE;
36                 swap(&result[j], &result[j + 1]);
37                 SAMPLE;
38                 no_change = 0;
39                 SAMPLE;
40             }
41             SAMPLE;
42         }
43         SAMPLE;
44     } while (!no_change);
45     SAMPLE;
46     return result;
47 }
48
49 int main()
50 {
51     int n = max;
52     double x[max];
53     srand(time(NULL));
54     for (int i = 0; i < n; i++)
55         x[i] = rand() % 100;
56     SAMPLE;
57     sort(x, n);
58     SAMPLE;
59     return 0;
60 }

```

Приложение В. Файл model.xml.

```

<model type = "Objects::AMC::Model" name = "lab4">
  <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>
<link type = "Objects::AMC::Link" name = "t1-->t2" probability = "1.0"
intensity = "2.51" deviation = "0.0" source = "t1" dest = "t2"></link>
<link type = "Objects::AMC::Link" name = "t2-->t3" probability = "1.0"
intensity = "20.95" deviation = "0.0" source = "t2" dest = "t3"></link>
<link type = "Objects::AMC::Link" name = "t3-->t4" probability = "1.0"
intensity = "115.66" deviation = "0.0" source = "t3" dest = "t4"></link>
<link type = "Objects::AMC::Link" name = "t4-->t5" probability = "1.0"
intensity = "0.0" deviation = "0.0" source = "t4" dest = "t5"></link>
<link type = "Objects::AMC::Link" name = "t5-->t6" probability = "1.0"
intensity = "0.18" deviation = "0.0" source = "t5" dest = "t6"></link>
<link type = "Objects::AMC::Link" name = "t6-->t7" probability = "0.9875"
intensity = "2.93" deviation = "0.0" source = "t6" dest = "t7"></link>
<link type = "Objects::AMC::Link" name = "t6-->t11" probability = "0.0125"
intensity = "2.22" deviation = "0.0" source = "t6" dest = "t11"></link>
<link type = "Objects::AMC::Link" name = "t7-->t8" probability = "0.3334"
intensity = "5.94" deviation = "0.0" source = "t7" dest = "t8"></link>
<link type = "Objects::AMC::Link" name = "t7-->t6" probability = "0.6666"
intensity = "0.0" deviation = "0.0" source = "t7" dest = "t6"></link>
<link type = "Objects::AMC::Link" name = "t8-->t9" probability = "1.0"
intensity = "14.0" deviation = "0.0" source = "t8" dest = "t9"></link>
<link type = "Objects::AMC::Link" name = "t9-->t10" probability = "1.0"
intensity = "15.58" deviation = "0.0" source = "t9" dest = "t10"></link>
<link type = "Objects::AMC::Link" name = "t10-->t6" probability = "1.0"
intensity = "0.0" deviation = "0.0" source = "t10" dest = "t6"></link>
<link type = "Objects::AMC::Link" name = "t11-->t5" probability = "0.9853"
intensity = "1.64" deviation = "0.0" source = "t11" dest = "t5"></link>
<link type = "Objects::AMC::Link" name = "t11-->t12" probability = "0.0147"
intensity = "1.68" deviation = "0.0" source = "t11" dest = "t12"></link>
</model>

```