

Программа расчёта сопротивлений между  
узлами цепи

Руководство программиста

Буланов Олег  
группа 301

16 февраля 2015 г.

## **Аннотация**

Данный программный документ содержит описание и характеристики программы "hw1.py". Программа рассчитывает сопротивления между всеми узлами электрической цепи. Для этого программа считывает xml представление графа сопротивлений результат в csv-файл.

## Содержание

<b>1</b>	<b>Назначение и условия применения программы</b>	<b>4</b>
1.1	Назначение программы . . . . .	4
1.2	Функции, выполняемые программой . . . . .	4
1.3	Условия, необходимые для выполнения программы . . . . .	4
1.3.1	Объём оперативной памяти . . . . .	4
1.3.2	Требования к составу и параметрам периферийных устройств	4
1.3.3	Требования к программному обеспечению . . . . .	4
<b>2</b>	<b>Характеристика программы</b>	<b>5</b>
2.1	Временные характеристики . . . . .	5
2.2	Режим работы программы . . . . .	5
2.3	Контроль правильности выполнения программы . . . . .	5
<b>3</b>	<b>Обращение к программе</b>	<b>5</b>
3.1	Запуск программы . . . . .	5
3.2	Выполнение программы . . . . .	5
3.3	Завершение работы программы . . . . .	5
<b>4</b>	<b>Входные и выходные данные</b>	<b>6</b>
4.1	Организация используемой входной информации . . . . .	6
4.2	Организация используемой выходной информации . . . . .	6
<b>5</b>	<b>Сообщения</b>	<b>6</b>

# **1. Назначение и условия применения программы**

## **1.1. Назначение программы**

При производстве печатных плат существует потребность в их тестировании и проверке. Одним из способов это сделать является измерение сопротивлений между разными узлами цепи и сравнение измеренных значений с эталонными сопротивлениями.

Для расчёта эталонных сопротивлений между всеми узлами цепь удобно представить в виде взвешенного ориентированного графа. Вершинами графа являются узлы электрической цепи, ребрами являются элементы электрической цепи, весами которых являются их сопротивления.

Программа предназначена для расчёта эталонных значений между всеми узлами электрической цепи, которая представлена в виде взвешенного графа сопротивлений. Граф должен быть представлен в виде xml-файла. Результатом выполнения программы будет csv-файл, в котором записаны значения эталонных сопротивлений между узлами.

## **1.2. Функции, выполняемые программой**

Основной функцией программы является загрузка графа сопротивлений из xml-файла и запись, полученной в результате работы программы матрицы, в csv-файл. Эта функция вызывается при помощи скрипта командной строки. В качестве аргументов командной строки программа принимает адрес xml-файла, из которого будет считан граф сопротивлений, и адрес csv-файла, в который будет записана матрица эталонных сопротивлений. В случае недостатка аргументов командной строки или отсутствия указанных файлов, программа завершает работу с ошибкой. Если программа успешно завершает свою работу, выводится время работы программы в миллисекундах.

## **1.3. Условия, необходимые для выполнения программы**

### **1.3.1. Объём оперативной памяти**

Программа требует около 7 Мб оперативной памяти, но это значение может меняться в зависимости от размера входных данных.

### **1.3.2. Требования к составу и параметрам периферийных устройств**

Никаких особых требований к составу и параметрам периферийных устройств программа "hw1.py" не предъявляет.

### **1.3.3. Требования к программному обеспечению**

Программа представляет список команд для интерпретатора python. Поэтому для запуска программы "hw1.py" необходимо, чтобы был установлен интерпретатор python

## 2. Характеристика программы

### 2.1. Временные характеристики

Программа "hw1.py" реализует алгоритм Флойда-Уоршела, который имеет кубическую скорость работы. То есть, если количество узлов в графе сопротивлений обозначить за  $n$ , то время работы алгоритма будет  $O(n^3)$ . Время в миллисекундах выводится в конце работы программы.

### 2.2. Режим работы программы

Программа "hw1.py" запускается только, когда её вызывают из командной строки с нужным количеством аргументов.

### 2.3. Контроль правильности выполнения программы

Проверить правильность работы программы можно проверить, например, введя недостаточное количество аргументов или некорректные значения аргументов. В этом случае программа выдаст сообщение об ошибке. Качество выходных данных можно проверить по следующему принципу. Если на основной диагонали нули, а вне вещественные числа, то скорее всего программа работает корректно.

## 3. Обращение к программе

### 3.1. Запуск программы

Чтобы запустить программу необходимо написать скрипт в командной строке и передать туда два аргумента (адрес xml-файла для считывания и адрес csv-файла для записи). Пример смотри ниже.

```
python hw1.py inpfle.xml outpfle.csv
```

### 3.2. Выполнение программы

Если количество переданных аргументов равно двум, то программа начинает работу. Если адрес xml-файла корректен, то программа считывает ребра и вершины графа из этого файла. Затем программа реализует алгоритм Флойда-Уоршила, используя считанные данные. Затем, если адрес csv-файла корректен, то программа записывает, полученную в результате работы алгоритма, матрицу в этот файл. Если аргументов недостаточно или адрес не корректен, программа завершает работу с ошибкой.

### 3.3. Завершение работы программы

Программа не имеет пользовательского интерфейса, поэтому завершает свою работу автоматически либо с ошибкой, если какие-нибудь данные были не корректны, либо успешно, сохраняя результат в csv-файл и выводя время своей работы.

## 4. Входные и выходные данные

### 4.1. Организация используемой входной информации

На вход в качестве аргументов командной строки программе подаётся адрес xml-файла, который хранит в себе представление графа сопротивлений, и адрес csv-файла, в который программа записывает матрицу сопротивлений графа.

### 4.2. Организация используемой выходной информации

Результатом работы программы служит csv-файл, адрес которого был предан в качестве аргумента командной строки, также программа выводит время своей работы в миллисекундах.

## 5. Сообщения

Программа может выдать три вида сообщений (См. Рис 1) в случаях:

1. В случае успешной работы, программа выдаёт время работы программы.
2. Если не достаточно, аргументов командной строки
3. Если адреса файлов не корректны

```
linux-lrw1:~/Oleg/hw1 # python3 hw1.py example_input_3_nodes.xml table.csv
working time: 54.800000000000004
linux-lrw1:~/Oleg/hw1 # 
linux-lrw1:~/Oleg/hw1 # python3 hw1.py example_input_3_nodes.xml
Error of using the program
usage: hw1.py <input_xml_file> <output_csv_file>
linux-lrw1:~/Oleg/hw1 # python3 hw1.py example_input_nodes.xml table.csv
Traceback (most recent call last):
  File "hw1.py", line 150, in <module>
    main(sys.argv[1:])
  File "hw1.py", line 144, in main
    floyd_warshall_from_xml_to_csv(argv[0], argv[1])
  File "hw1.py", line 133, in floyd_warshall_from_xml_to_csv
    tree = ElementTree.parse(xml_address)
  File "/usr/lib/python3.4/xml/etree/ElementTree.py", line 1187, in parse
    tree.parse(source, parser)
  File "/usr/lib/python3.4/xml/etree/ElementTree.py", line 587, in parse
    source = open(source, "rb")
FileNotFoundError: [Errno 2] No such file or directory: 'example_input_nodes.xml'
```

Рис. 1: Три вида сообщений соответственно для трех случаев

[illegible]