

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Омский государственный технический университет»
Кафедра «Информатика и вычислительная техника»

Отчёт по лабораторной работе № 9
по дисциплине
«Проектирование и тестирование программного обеспечения»

Выполнил:
Студент гр. ПИН-211
Сероухов Е.С. _____
(подп., дата)

Проверил:
Старший преподаватель каф. ИВТ
Карабцов Р.Д. _____
(подп., дата)

ВВЕДЕНИЕ

Цель работы: научиться применять алгоритмы обхода графов. Решить поставленную задачу.

Задача работы: Задача 9.3 (Лесенки редактирования)

PC/UVa IDs: 110905/10029

Назовем ступенькой редактирования (edit step) такое преобразование слова x в слово y , что слова x и y принадлежат словарю и слово x может быть преобразовано в слово y путем добавления, удаления или изменения одной буквы. Преобразования dig в dog и dog в do являются ступеньками редактирования. Лесенка ступенек редактирования (edit step ladder) - это отсортированная в описанном порядке последовательность слов. Для заданного словаря вы должны найти самую большую лесенку ступенек редактирования.

Входные данные

Входные данные состоят из словаря: набора слов в нижнем регистре в лексикографическом порядке по одному слову на строку. Длина каждого слова не превышает 16 букв, и в словаре не более 25 000 слов.

Выходные данные

Выходные данные состоят из одного целого числа, количества слов в самой большой лесенке ступенек редактирования.

Пример входных данных

cat
dig
dog
fig
fin
fine
fog
log
wine

Соответствующие выходные данные

5

РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

1. Для решения задачи был написан код на ЯП Java в среде IntelliJ IDEA 2022.1.2.

Входные данные загружаются из файла “data.txt” (рисунок 1).

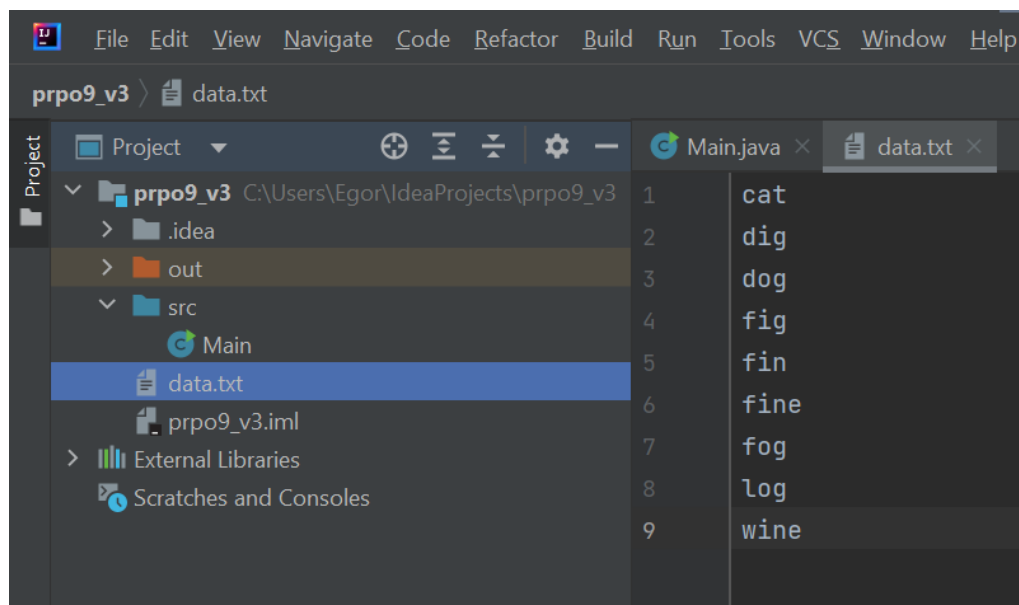


Рисунок 1 – Входные данные

Код программы:

```
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Scanner;

public class Main {
    final static List<String> Data = new ArrayList<>();
    public static void dataUploading(String path) {
        try (FileReader fr= new FileReader(path)){
            Scanner scan = new Scanner(fr);
            while (scan.hasNextLine())
                Data.add(scan.nextLine());
        }
        catch(IOException ex){
            System.out.println(ex.getMessage());
        }
    }
    public static void printHashMap(HashMap<String, Integer> hashMap) {
        for (String key : hashMap.keySet())
            System.out.println(key + ": " + hashMap.get(key));
    }

    public static void main(String[] args) {
        HashMap<String, Integer> L = new HashMap<>();
        int maxLength = 0;
```

```

String file = "data.txt";
dataUploading(file);
System.out.println("Входные данные загружены из файла: " + file);

for (String word : Data){
    int length = 1;

    // Insert c before word[i].
    for (int i = 0; i <= word.length(); ++i)
        for (char c = 'a'; c <= 'z'; ++c) {
            StringBuilder wordNew = new StringBuilder(word);
            wordNew.insert(i, c);
            if (wordNew.toString().compareTo(word) > 0)
                break;
            if (L.containsKey(wordNew.toString()))
                length = Math.max(length, L.get(wordNew.toString()) +
1);
        }

    // Delete word[i].
    for (int i = 0; i < word.length(); ++i) {
        StringBuilder wordNew = new StringBuilder(word);
        wordNew.deleteCharAt(i);
        if (L.containsKey(wordNew.toString()))
            length = Math.max(length, L.get(wordNew.toString()) + 1);
    }

    // Change word[i].
    for (int i = 0; i < word.length(); ++i)
        for (char c = 'a'; c <= 'z' && c != word.charAt(i); ++c) {
            StringBuilder wordNew = new StringBuilder(word);
            wordNew.setCharAt(i, c);
            if (wordNew.toString().compareTo(word) > 0)
                break;
            if (L.containsKey(wordNew.toString()))
                length = Math.max(length, L.get(wordNew.toString()) +
1);
        }

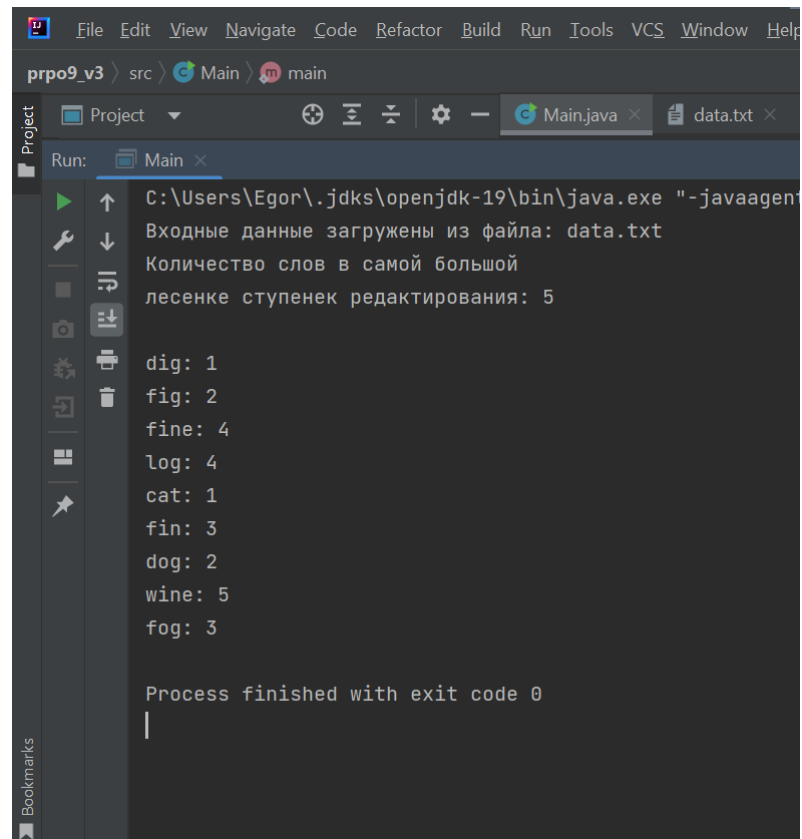
    L.put(word, length);

    maxLength = Math.max(maxLength, length);
}

System.out.println("Количество слов в самой большой \n" +
    "лесенке ступенек редактирования: "+maxLength+"\n");
printHashMap(L);
}
}

```

2. Было проведено тестирование работы программы (рисунок 2).



```
C:\Users\Egor\.jdk\openjdk-19\bin\java.exe "-javaagent
Входные данные загружены из файла: data.txt
Количество слов в самой большой
лесенке ступенек редактирования: 5

dig: 1
fig: 2
fine: 4
log: 4
cat: 1
fin: 3
dog: 2
wine: 5
fog: 3

Process finished with exit code 0
|
```

Рисунок 2 – Тестирование работы программы.

ЗАКЛЮЧЕНИЕ

Вывод: в ходе работы была написана программа на ЯП Java. Программа решила поставленную задачу и прошла проверку при тестировании.