|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ | | | | | | | |  |
|  | | | |  | | |  | | |
|  | | | ФГБОУ ВО «Пермский государственный национальный исследовательский университет» | | | | |  | |
|  | | | |  | | |  | | |
|  | | ОТЧЕТ  по лабораторной работе «Сравнение парадигм программирования»  Вариант 3 | | | | | |  | |
|  | | | |  | | |  | | |
|  | Работу выполнил  Студент гр.ПМИ-1  Халдина Ю. А. \_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 | | | |  | Проверил  Кандидат технических наук кафедры МОВС  Городилов А. Ю.\_\_\_\_\_\_\_\_\_\_  «\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 | | |  |
|  |  | | | |  |  | | |  |
| Пермь 2020 | | | | | | | | | |
| СОДЕРЖАНИЕ  [1 Постановка задачи 3](#_Toc38239574)  [2 Описание работы алгоритма 4](#_Toc38239575)  [3 Сравнение языков 5](#_Toc38239576)  [Трудоёмкость разработки 5](#_Toc38239577)  [Трудоёмкость тестирования и отладки 5](#_Toc38239578)  [Объём полученного кода 5](#_Toc38239579)  [Быстродействие и скорость работы 5](#_Toc38239580) | | | | | | | | | |

# Постановка задачи

Написать две компьютерные программы (в императивном стиле (C++, C# или Java) и в декларативном стиле (Haskell, F# или Lisp)), реализующие выбранный алгоритм. Проверок на корректность ввода не требуется (кроме явно указанных в задании). Подготовить набор тестов (минимум 5) с разным объемом входных данных. Должна быть возможность быстро запустить программу для различных тестов, не набирая с клавиатуры все входные данные. Протестировать программу на наборе тестов и отладить ее. Сравнить скорость работы программ при разных объемах входных данных.

**Задача**: Ориентированный граф задан списком ребер. Каждое ребро представляет собой тройку (вершина, вершина, длина). Все длины являются натуральными числами. Найти расстояние между двумя заданными вершинами в графе. Если пути между указанными вершинами нет, вывести -1.

# Описание работы алгоритма

Так как дан взвешенный ориентированный граф, веса в котором являются натуральными числами (следовательно, веса не могут быть отрицательны), для решения задачи был выбран алгоритм Дейкстры.

Для реализации алгоритма были выбраны языки C# и F#.

Алгоритм принимает на вход начальную вершину и список рёбер, представленных в виде (вершина, вершина, длина ребра).

* В расстояние от начальной вершины до каждой вершины, отличной от начальной, присваивается бесконечность (в нашем случае большое число).
* Пока остались непосещённые вершины:
* Производится поиск такой непосещённой вершины A, что расстояние от начальной вершины до этой вершины минимально.
* Эта вершина становится посещённой.
* Для всех непосещённых вершин B, которые соединены ребром с A:
* Если расстояние от начальной вершины до B больше суммы расстояния до вершины А и длины ребра, то расстоянию до вершины В присваивается эта сумма.

# Сравнение языков

## Трудоёмкость разработки

В силу того, что на императивном языке писать более привычно, написание алгоритма на C# было намного легче, чем на F#.

## Трудоёмкость тестирования и отладки

Обе программы были созданы в Visual Studio 2019, поэтому отладка была одинаковой.

Тестирование также не имело особых различий. Больше ошибок было выявлено при тестировании программы на F#.

## Объём полученного кода

Алгоритм на F# занимает 32 строки кода и 1191 символов без пробельных.

Алгоритм на C# занимает 74 строки кода (из них просто с фигурными скобками 23) и 1875 символов без пробельных.

## Быстродействие и скорость работы

Замер времени выполнения каждого теста в обеих программах производился с помощью объекта класса Stopwatch. Результаты замеров представлены в таблице 3.1.

Таблица 3.1 — Результаты замеров

|  |  |  |
| --- | --- | --- |
|  | C# | F# |
| Простой граф | 0.0000163 | 0.0000796 |
| Маленький граф с циклом | 0.0000063 | 0.0000147 |
| Маленький граф с большими числами | 0.0000157 | 0.0000600 |
| Пустой граф | 0.0000015 | 0.0000067 |
| Граф с повторяющимися рёбрами | 0.0002721 | 0.0018596 |
| Большой граф | 0.0015176 | 0.0082840 |