

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7.2**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема. Нелинейные структуры

Графы: создание, алгоритмы обхода, важные задачи теории

графов

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-43-23 | Кощеев М. И. |
| Принял старший преподаватель | Рысин М.Л. |

Москва 2024

**СОДЕРЖАНИЕ**

[1 ЦЕЛЬ 3](#_Toc184123958)

[2 ХОД РАБОТЫ 4](#_Toc184123959)

[**4** **ВЫВОДЫ** 9](#_Toc184123960)

[**5** **ИСПОЛЬЗУЕМЫЙ ИСТОЧНИК** 10](#_Toc184123961)

# ЦЕЛЬ

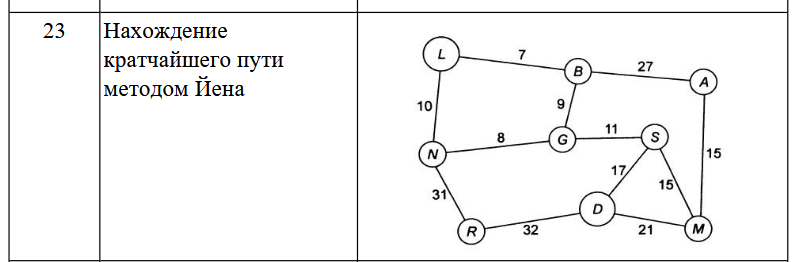
Освоить приемы создания, алгоритмы обхода, важные задачи теории графов.

# 2 ХОД РАБОТЫ

Составить программу создания графа и реализовать процедуру для работы с

графом, определенную индивидуальным вариантом задания.

Вариант 23:



**Алгоритм (dijkstra):**

1. Создаём структуры для хранения расстояний, предыдущих вершин и посещённых вершин.
2. Инициализируем все расстояния как бесконечность, кроме начальной вершины (её расстояние равно 0).
3. Создаём приоритетную очередь для хранения вершин с минимальными расстояниями и добавляем в неё начальную вершину.
4. Пока очередь не пуста:

a) Извлекаем вершину с минимальным расстоянием.

b) Если вершина уже была посещена, пропускаем её.

c) Отмечаем вершину как посещённую.

d) Для всех соседей текущей вершины: i. Если ребро между вершинами заблокировано, пропускаем его. i. Вычисляем новое расстояние до соседа. i. Если оно меньше текущего, обновляем расстояние и добавляем соседа в очередь.

1. Если достигли конечной вершины, восстанавливаем путь от конца до начала, используя предыдущие вершины.

Код программы



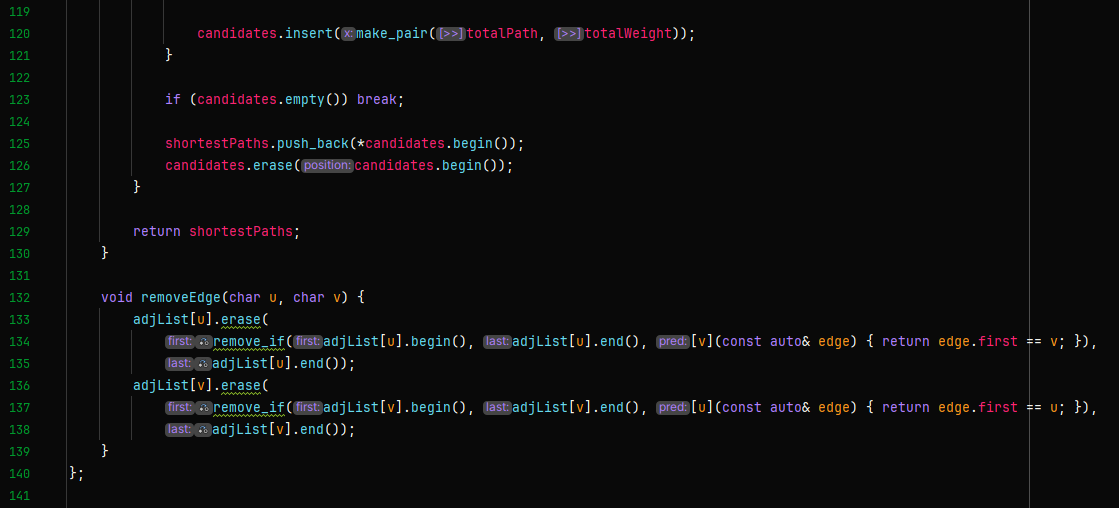


**Алгоритм (yenKShortestPaths):**

1. Получаем первый кратчайший путь с помощью алгоритма Дейкстры.
2. Добавляем его в список кратчайших путей.
3. Для каждого следующего пути (от 2-го до k-го): a) Для каждого ребра в последнем найденном пути: i. Удаляем это ребро из графа, блокируем его. i. Запускаем алгоритм Дейкстры с заблокированными ребрами. i. Если найден новый путь, вычисляем его длину и добавляем в список кандидатов. b) Из всех возможных путей выбираем тот, который имеет минимальную длину.
4. Повторяем, пока не найдём k путей или не исчерпаем все возможные пути.

Код

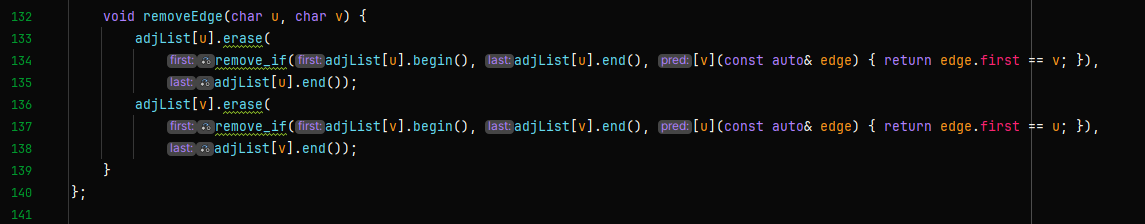




**Алгоритм (removeEdge):**

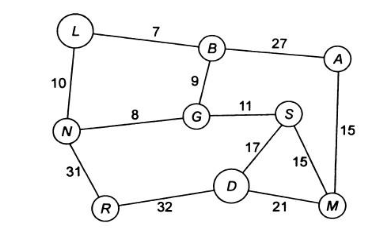
1. Для каждой вершины находим ребро с указанным соседним узлом и удаляем его.
2. Повторяем для второй вершины, удаляя обратное ребро.

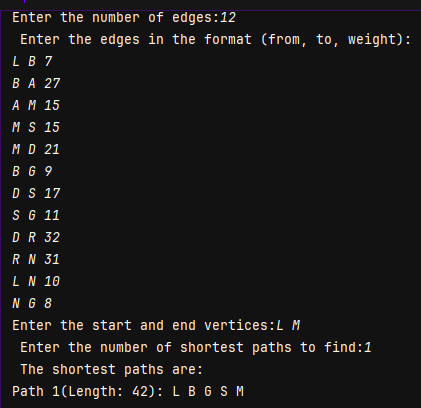
Код



Тестирование

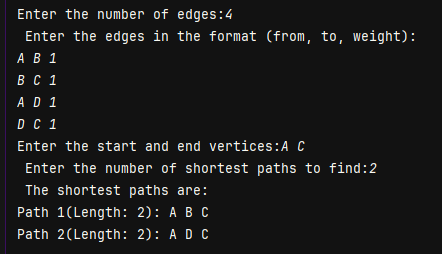
1. Граф из варианта





1. Произвольный граф





1. **ВЫВОДЫ**

В результате проделанной работы были освоены приемы создания, алгоритмы обхода, важные задачи теории графов.

1. **ИСПОЛЬЗУЕМЫЙ ИСТОЧНИК**

1. М.Л. РЫСИН, М.В. САРТАКОВ, М.Б. ТУМАНОВА Учебно-методическое

пособие СиАОД часть 2