МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7.2

по дисциплине

«Структуры и алгоритмы обработки данных»

Тема «Графы: создание, алгоритмы обхода, важные задачи теории графов»

Выполнил студент группы ИКБО-13-23 Русаков М.Ю.

Принял преподаватель Сорокин А.В.

Москва 2024 г.

Оглавление

Цель работы: изучить важные задачи теории графов

1. Выполнение работы

1.1. Формулировка задачи

Реализовать построение минимального остовного дерева при помощи алгоритма Прима.

1.2. Математическая модель

Алгоритм Прима предназначен для построения минимального остовного дерева T связного взвешенного графа $G = (V, E)$, где V — множество вершин, E — множество рёбер с весами $w : E \to R^+$. Алгоритм начинается с произвольной вершины $v_0 \in V$, которой присваивается ключевой вес $\text{key}[v] = 0$. Для всех остальных вершин $v \in V \setminus \{v_0\}$ их ключевые веса инициализируются как $\text{key}[v] = \infty$. На каждом шаге алгоритм выбирает вершину $v \in V \setminus T$, обладающую минимальным ключевым весом. Эта вершина добавляется в остовное дерево T.

После добавления вершины u, для каждой соседней вершины v, связанной с u ребром $e = (u, v) \in E$, проверяется условие:

$$w(u, v) < \text{key}[v]$$

Если оно выполняется, то ключевой вес вершины обновляется:

$$\text{key}[v] = w(u, v),$$

а вершина u записывается как родитель вершины v в дереве:

$$\text{parent}[v] = u$$

Процесс повторяется $|V| - 1$ раз, пока все вершины не будут включены в дерево T.

В результате получается минимальное остовное дерево.

1.3. Код программы

Приведем код, реализующий алгоритм Прима на языке программирования C++, ниже

```
#include <iostream>
#include <vector>
#include <limits.h>

using namespace std;

```

```cpp
int min_key(
    const vector<int>& key,
    const vector<bool>& included,
    int vertices
) {
    int min = INT_MAX, minIndex;

    for (int v = 0; v < vertices; v++) {
        if (!included[v] && key[v] < min) {
            min = key[v];
            minIndex = v;
        }
    }

    return minIndex;
}

void print_graph(
    const vector<int>& parent,
    const vector<vector<int>>& graph,
    int vertices
) {
    cout << "Ребро \tВес\n";
    for (int i = 1; i < vertices; i++) {
        cout << parent[i] + 1 << " - " << i + 1 << "\t" << graph[i]
[parent[i]] << "\n";
    }
}

void find_prim_mst(const vector<vector<int>>& graph, int vertices) {
    vector<int> parent(vertices);
    vector<int> key(vertices, INT_MAX);
    vector<bool> included(vertices, false);

    key[0] = 0;
    parent[0] = -1;

    for (int count = 0; count < vertices - 1; count++) {
        int u = min_key(key, included, vertices);
        included[u] = true;

        for (int v = 0; v < vertices; v++) {
            if (graph[u][v] && !included[v] && graph[u][v] < key[v]) {
                parent[v] = u;
                key[v] = graph[u][v];
            }
        }
    }

    print_graph(parent, graph, vertices);
}

int main() {
```

```
59      int vertices, edges;
60      cout << "Введите количество вершин и ребер: ";
61      cin >> vertices >> edges;
62
63      vector<vector<int>> graph(vertices, vector<int>(vertices, 0));
64
65       cout << "Введите ребра в следующем формате: вершина1 вершина2
    вес\n";
66      for (int i = 0; i < edges; i++) {
67          int u, v, weight;
68          cin >> u >> v >> weight;
69
70          u--;
71          v--;
72
73          graph[u][v] = weight;
74          graph[v][u] = weight;
75      }
76
77      cout << "\nМинимальное остовное дерево (алгоритм Прима):\n";
78      find_prim_mst(graph, vertices);
79
80      return 0;
81  }
```

1.4. Тестирование

1.4.1. Тест №1

В качестве теста №1 приведен данный в формулировке задания граф (см. рис. 1).

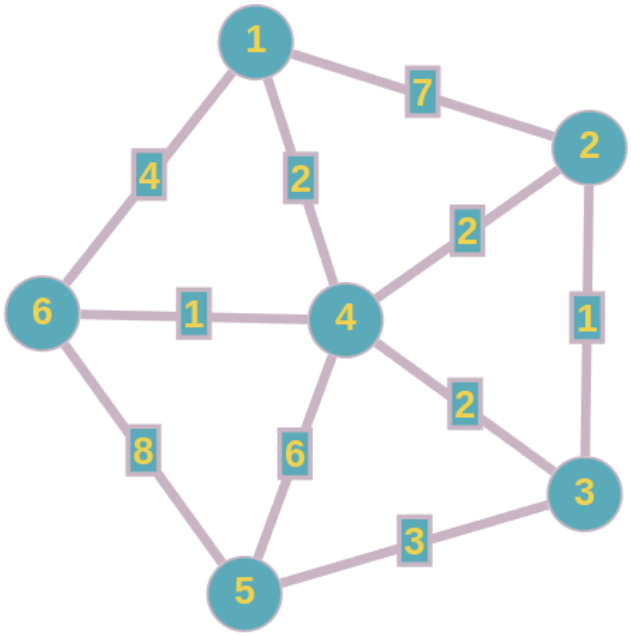

Рисунок 1 – Граф №1

В результате работы программы получается следующий вывод (см. рис. 2)

```
Введите количество вершин и ребер: 6 10
Введите ребра в следующем формате: вершина1 вершина2 вес
1 2 7
1 4 2
1 6 4
2 4 2
2 3 1
3 4 2
3 5 3
5 4 6
5 6 8
4 6 1

Минимальное остовное дерево (алгоритм Прима):
Ребро   Вес
4 - 2   2
2 - 3   1
1 - 4   2
3 - 5   3
4 - 6   1
```

Рисунок 2 – Вывод программы для входных данных теста №1

Вывод соответствует следующему минимальному остовному дереву (см. рис. 3)

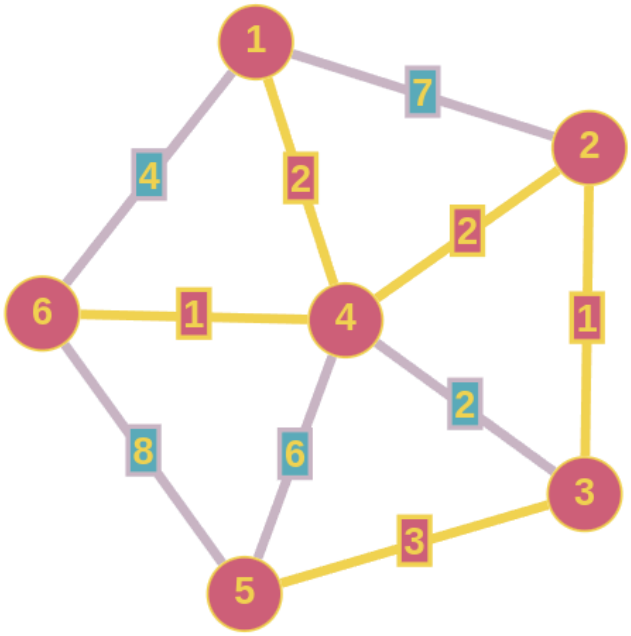

Рисунок 3 – Граф №1

1.4.2. Тест №2

В качестве теста №2 приведем следующий граф (см. рис. 4).

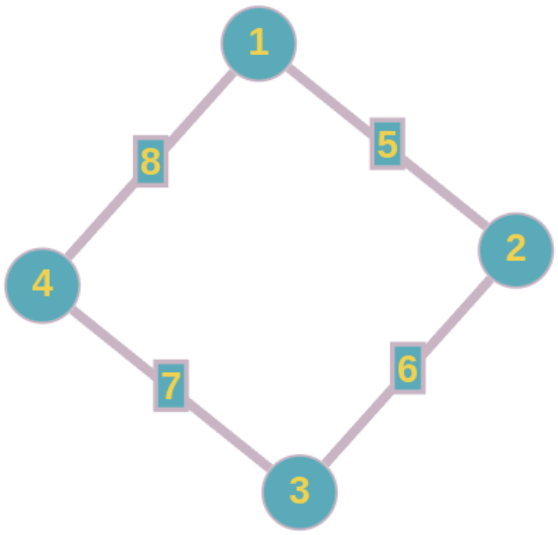

Рисунок 4 – Граф №2

В результате работы программы получается следующий вывод (см. рис. 5)

```
Введите количество вершин и ребер: 4 4
Введите ребра в следующем формате: вершина1 вершина2 вес
1 2 5
1 4 8
2 3 6
3 4 7

Минимальное остовное дерево (алгоритм Прима):
Ребро   Вес
1 - 2   5
2 - 3   6
3 - 4   7
```

Рисунок 5 – Вывод программы для входных данных теста №2

Вывод соответствует следующему минимальному остовному дереву (см. рис. 6)

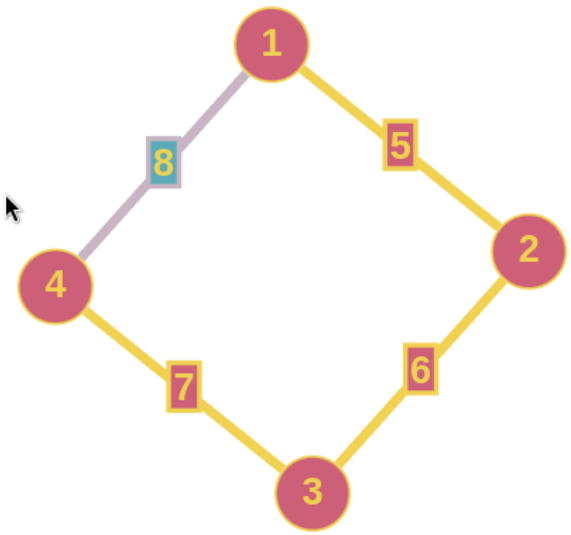

Рисунок 6 – Граф №1

1.4.3. Тест №3

В качестве теста №2 приведем следующий граф, уже являющийся деревом, веса всех ребер которого равны 1 (см. рис. 4).

Рисунок 7 – Граф №3

В результате работы программы получается следующий вывод (см. рис. 5). Как видно из полученного результата, мы получили исходный граф.

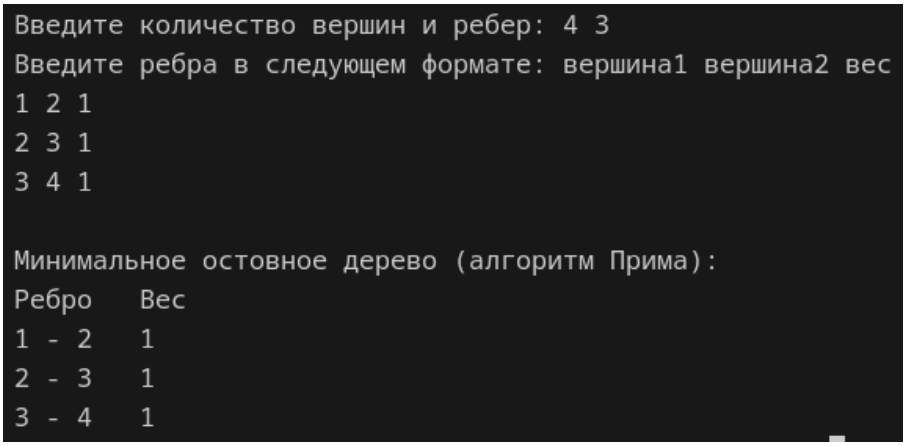

Рисунок 8 – Вывод программы для входных данных теста №3

2. Вывод

В ходе выполнения практической работы был изучен алгоритм Прима для построения минимального остовного дерева связного взвешенного графа. Были исследованы теоретические основы алгоритма, включая его жадный подход к выбору рёбер с минимальным весом на каждом шаге.

В процессе работы была реализована программа на языке C++, позволяющая находить минимальное остовное дерево для произвольного графа. В качестве метода хранения графа в памяти была использована матрица смежности. Программа предусматривает ввод графа с клавиатуры в формате списка рёбер, что обеспечивает её универсальность и возможность работы с различными графами.

Были проведены тестовые прогоны программы на 3-х заданных графах, в результате чего были получены минимальные остовные деревья. Результаты подтвердили корректность реализации и соответствие алгоритма его теоретической модели.