

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**
(БГТУ им. В.Г. Шухова)



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №4.5

по дисциплине: Дискретная математика

тема: «Кратчайшие пути между каждой парой вершин
во взвешенном орграфе»

Выполнил: ст. группы ПВ-221
Лоёк Никита Викторович

Проверили:
Бондаренко Татьяна Владимировна
Рязанов Юрий Дмитриевич

Белгород 2023 г.

Лабораторная работа № 4.5

Цель работы: изучить алгоритмы нахождения кратчайших путей между каждой парой вершин во взвешенном орграфе, научиться использовать их при решении различных задач.

Задания

1. Изучить алгоритмы нахождения кратчайших путей между каждой парой вершин во взвешенном орграфе.

```
bool IsAll(vector<bool> &v) {
    bool isAll = true;

    for (int i = 0; i < v.size() && isAll; ++i) {
        isAll = isAll && v[i];
    }

    return isAll;
}

vector<int> GetAdjacent(vector<vector<int>> &matrix, int v) {
    vector<int> adjacentVertexes;

    for (int i = 0; i < matrix.size(); ++i) {
        if (matrix[v - 1][i] > 0) {
            adjacentVertexes.push_back(i + 1);
        }
    }

    return adjacentVertexes;
}

vector<int> GetMinWays(vector<vector<int>> &matrix, int vStart) {
    vector<int> t(matrix.size(), -1);
    vector<int> d(matrix.size(), -1);
    vector<bool> v(matrix.size(), false);

    t[vStart - 1] = 0;
    v[vStart - 1] = true;
    d[vStart - 1] = 0;

    int nextYw = 0;
    int nextY = vStart;
    while (!IsAll(v) && nextYw != -1) {
        vector<int> adjacent = GetAdjacent(matrix, vStart);
        nextYw = -1;

        for (int i = 0; i < adjacent.size(); ++i) {
            if (d[adjacent[i] - 1] == -1) {
                d[adjacent[i] - 1] = d[vStart - 1] + matrix[vStart - 1][adjacent[i] - 1];
                t[adjacent[i] - 1] = vStart;
            } else {
                if (d[adjacent[i] - 1] > d[vStart - 1] + matrix[vStart - 1][adjacent[i] - 1]) {
                    d[adjacent[i] - 1] = d[vStart - 1] + matrix[vStart - 1][adjacent[i] - 1];
                    t[adjacent[i] - 1] = vStart;
                }
            }
        }

        for (int i = 0; i < d.size(); ++i) {
            if (d[i] != -1 && !v[i]) {
                if (nextYw == -1) {
                    nextYw = d[i];
                    nextY = i + 1;
                } else {
                    if (d[i] < nextYw) {

```

```

        nextYw = d[i];
        nextY = i + 1;
    }
}

vStart = nextY;
if (vStart != -1) {
    v[vStart - 1] = true;
}

return t;
}

int TreeToLen(vector<vector<int>> &matrix, vector<int> &tree, int start, int end) {
    int sumLen = 0;

    while (end != start && end != -1) {
        if (tree[end - 1] != -1) {
            sumLen += matrix[tree[end - 1] - 1][end - 1];
        }
        end = tree[end - 1];
    }

    if (end == -1) {
        return -1;
    } else {
        return sumLen;
    }
}

vector<vector<int>> GetAllPairsMinWays(vector<vector<int>> &matrix) {
    vector<vector<int>> minWays(matrix.size(), vector<int>(matrix.size(), 0));

    for (int i = 0; i < matrix.size(); ++i) {
        vector<int> tree = GetMinWays(matrix, i + 1);

        for (int j = 0; j < minWays.size(); ++j) {
            minWays[i][j] = TreeToLen(matrix, tree, i + 1, j + 1);
        }
    }

    return minWays;
}

```

2. Разработать и реализовать алгоритм решения задачи (см. варианты заданий).

Во взвешенном орграфе найти все такие вершины v_i , что сумма кратчайших расстояний от всех вершин орграфа до v_i меньше суммы кратчайших расстояний от v_i до всех вершин орграфа.

```
vector<int> GetLessInThanOut(vector<vector<int>> &matrix) {
    vector<int> vertexes;

    vector<vector<int>> allPairsMinWays = GetAllPairsMinWays(matrix);

    for (int i = 0; i < allPairsMinWays.size(); ++i) {
        int sumInVertex = 0;
        int sumOutVertex = 0;
        for (int j = 0; j < allPairsMinWays.size(); ++j) {
            if (allPairsMinWays[j][i] != -1) {
                sumInVertex += allPairsMinWays[j][i];
            }

            if (allPairsMinWays[i][j] != -1) {
                sumOutVertex += allPairsMinWays[i][j];
            }
        }

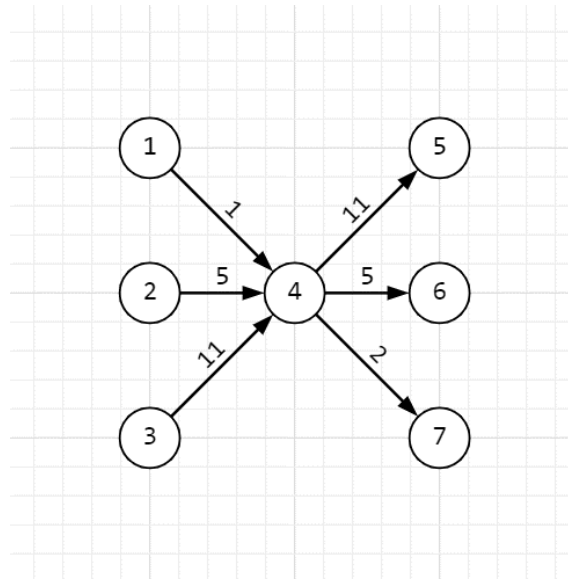
        if (sumInVertex < sumOutVertex) {
            vertexes.push_back(i + 1);
        }
    }

    return vertexes;
}
```

3. Подобрать тестовые данные. Результат представить в виде диаграммы графа.

Тест 1:

Изначальный Граф:



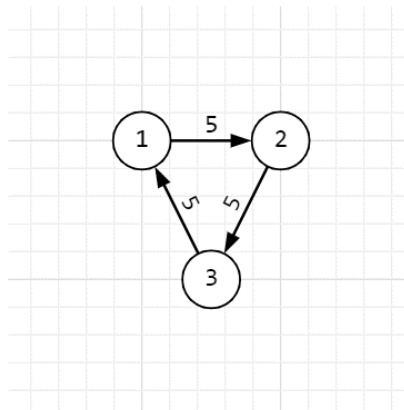
Результат работы программы:

```
0 0 0 1 0 0 0
0 0 0 5 0 0 0
0 0 0 11 0 0 0
0 0 0 0 11 5 2
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 2 3 4
```

Process finished with exit code 0

Тест 2:

Изначальный Граф:



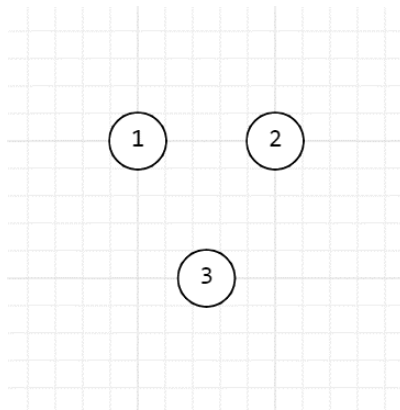
Результат работы программы:

```
0 5 0
0 0 5
5 0 0
```

Process finished with exit code 0

Тест 3:

Изначальный Граф:



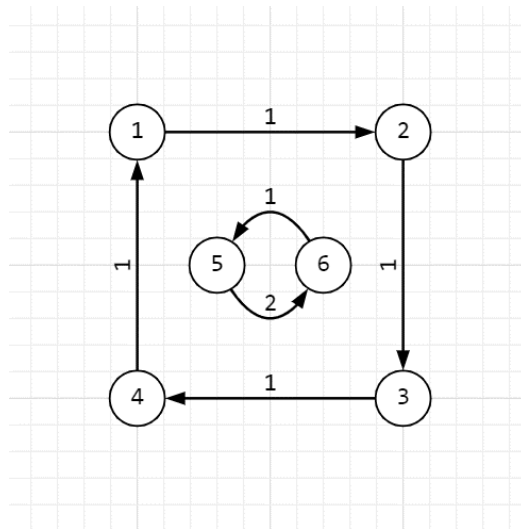
Результат работы программы:

```
0 0 0
0 0 0
0 0 0
```

Process finished with exit code 0

Тест 4:

Изначальный Граф:



Результат работы программы:

```
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
1 0 0 0 0 0
0 0 0 0 0 2
0 0 0 0 1 0
5
```

Process finished with exit code 0

Вывод

Вывод: в ходе работы я изучил алгоритм Дейкстры для нахождения кратчайших путей между вершинами взвешенного орграфа, научился рационально использовать его при решении различных задач.