

РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №3.4
по дисциплине: Дискретная математика
тема: «Упорядоченные множества»

Выполнил: ст. группы ПВ-211
Чувилко Илья Романович

Проверили:
Рязанов Юрий Дмитриевич
Бондаренко Татьяна Владимировна

Белгород 2022 г.

Вариант №9

Цель работы: изучить упорядоченные множества, алгоритм топологической сортировки, научиться представлять множества диаграммами Хассе, находить минимальные (максимальные) и наименьшие (наибольшие) элементы упорядоченного множества.

1. Написать программы, формирующие матрицы отношений в соответствии с вариантом задания (табл. 5), на множествах M1 и M2.

Условие: $A = \{(a, b) \mid a_x < b_x \text{ и } a_y \leq b_y\}$

Код:

```
#include <iostream>
#include <vector>
#include "algorithm"

#include "BinaryRelations/BinaryRelations.h"

using namespace std;

binaryRelation
getRelationFromPoints(const vector<pair<int, int>> &points) {
    auto dimension = points.size();
    binaryRelation res(dimension, binaryRelationMatrixRow(dimension, false));
    for (int a = 1; a < dimension; ++a) {
        for (int b = 1; b < dimension; ++b) {
            if (points[a].first < points[b].first &&
                points[a].second <= points[b].second) {
                res[a][b] = true;
            }
        }
    }
    return res;
}

int main() {
    vector<pair<int, int>> M1 = {{INT32_MIN, INT32_MIN},
                                {-1, -1}, {-1, 0}, {-1, 1},
                                {0, -1}, {0, 0}, {0, 1},
                                {1, -1}, {1, 0}, {1, 1}};
    vector<pair<int, int>> M2 = {{INT32_MIN, INT32_MIN},
                                {-2, 0},
                                {-1, -1}, {-1, 0},
                                {-1, 1}, {0, -2}, {0, -1}, {0, 0}, {0, 1}, {0, 2},
                                {1, -1}, {1, 0}, {1, 1},
                                {2, 0}};

    binaryRelation relationOnM1 = getRelationFromPoints(M1);
    binaryRelation relationOnM2 = getRelationFromPoints(M2);

    cout << "m1:\n";
    binaryRelation_output(relationOnM1);
    cout << "\nm2:\n";
    binaryRelation_output(relationOnM2);

    return 0;
}
```

Получим матрицы отношений на множествах M1 и M2:

Relation On M1:	Relation On M2:
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1	0 0 0 1 1 0 0 1 1 1 0 1 1 1
0 0 0 0 0 1 1 0 1 1	0 0 0 0 0 0 1 1 1 1 1 1 1 1
0 0 0 0 0 0 1 0 0 1	0 0 0 0 0 0 0 0 1 1 1 0 1 1
0 0 0 0 0 0 0 1 1 1	0 0 0 0 0 0 0 0 0 1 1 0 0 1
0 0 0 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0

2. Написать программы, формирующие матрицы отношения доминирования по матрицам отношения порядка.

```

binaryRelation
binaryRelation_getDominanceRelation(const binaryRelation &relation) {
    if (!binaryRelation_isOrderBR(relation))
        throw std::runtime_error("Binary relation must be a relation of order");
    auto dimension = relation.size();
    auto res = relation;
    for (int i = 1; i < dimension; ++i)
        res[i][i] = false;
    for (int z = 1; z < dimension; ++z)
        for (int x = 1; x < dimension; ++x)
            for (int y = 1; y < dimension; ++y)
                if (res[x][z] && res[z][y])
                    res[x][y] = false;
    return res;
}

```

```

int main() {
    vector<pair<int, int>> M1 = {{INT32_MIN, INT32_MIN},
        {-1, -1}, {-1, 0}, {-1, 1},
        {0, -1}, {0, 0}, {0, 1},
        {1, -1}, {1, 0}, {1, 1}};
    vector<pair<int, int>> M2 = {{INT32_MIN, INT32_MIN},
        {-2, 0},
        {-1, -1}, {-1, 0},
        {-1, 1}, {0, -2}, {0, -1}, {0, 0}, {0, 1}, {0, 2},
        {1, -1}, {1, 0}, {1, 1},
        {2, 0}};
}

```

```

binaryRelation relationOnM1 = getRelationFromPoints(M1);
binaryRelation relationOnM2 = getRelationFromPoints(M2);

```

```

binaryRelation relationOnM1Dominance = binaryRelation_getDominanceRelation(relationOnM1);
binaryRelation relationOnM2Dominance = binaryRelation_getDominanceRelation(relationOnM2);

```

```

cout << "m1:\n";
binaryRelation_output(relationOnM1Dominance);
cout << "\nm2:\n";
binaryRelation_output(relationOnM2Dominance);

return 0;
}

```

Получим матрицы отношений доминирования на множествах M1 и M2:

Relation On M1 Dominance:	Relation On M2 Dominance:
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0	0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0	0 0 0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 0 1 1 0 0 0	0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 1 1 1	0 0 0 0 0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1
	0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0 0 0 0 0 0 0

3. Написать программу, реализующую алгоритм топологической сортировки по матрице отношения доминирования

```

vector<vector<int>>
binaryRelation_topologicalSort(const binaryRelation &relation) {
    binaryRelation dominanceRelation =
        binaryRelation_getDominanceRelation(relation);
    auto dimension = relation.size();

    vector<int> W(dimension, 0);
    for (int a = 1; a < dimension; ++a)
        for (int b = 1; b < dimension; ++b)
            W[a] += dominanceRelation[b][a];
    std::vector<std::vector<int>> levels;
    while (std::find_if(W.begin() + 1, W.end(),
        [&](const auto &item) {
            return item >= 0;
        }) != W.end()) {
        levels.emplace_back();
        for (int i = 1; i < dimension; ++i) {
            if (!W[i]) {
                levels.back().push_back(i);
                W[i] = -1;
            }
        }
        for (const auto &a: levels.back())
            for (int b = 1; b < dimension; ++b)
                W[b] -= dominanceRelation[a][b];
    }
}

```

```

    return levels;
}

void outputVector(vector<pair<int, int>> m, vector<vector<int>> &a) {
    for (auto &i : a) {
        for (int j : i) {
            cout << "(" << m[j].first << "; " << m[j].second << ") ";
        }
        cout << '\n';
    }
}

int main() {
    vector<pair<int, int>> M1 = {{INT32_MIN, INT32_MIN},
                                {-1, -1}, {-1, 0}, {-1, 1},
                                {0, -1}, {0, 0}, {0, 1},
                                {1, -1}, {1, 0}, {1, 1}};
    vector<pair<int, int>> M2 = {{INT32_MIN, INT32_MIN},
                                {-2, 0},
                                {-1, -1}, {-1, 0},
                                {-1, 1}, {0, -2}, {0, -1}, {0, 0}, {0, 1}, {0, 2},
                                {1, -1}, {1, 0}, {1, 1},
                                {2, 0}};

    auto relationOnM1 = getRelationFromPoints(M1);
    auto relationOnM2 = getRelationFromPoints(M2);

    auto M1Levels = binaryRelation_topologicalSort(relationOnM1);
    auto M2Levels = binaryRelation_topologicalSort(relationOnM2);

    cout << "Relation On M1 Topological Sort:\n";
    outputVector(M1, M1Levels);
    cout << "\nRelation On M2 Topological Sort:\n";
    outputVector(M2, M2Levels);

    return 0;
}

```

Получим распределение элементов по уровням на множествах M1 и M2:

Relation On M1 Topological Sort:

```

(-1; -1) (-1; 0) (-1; 1)
(0; -1) (0; 0) (0; 1)
(1; -1) (1; 0) (1; 1)

```

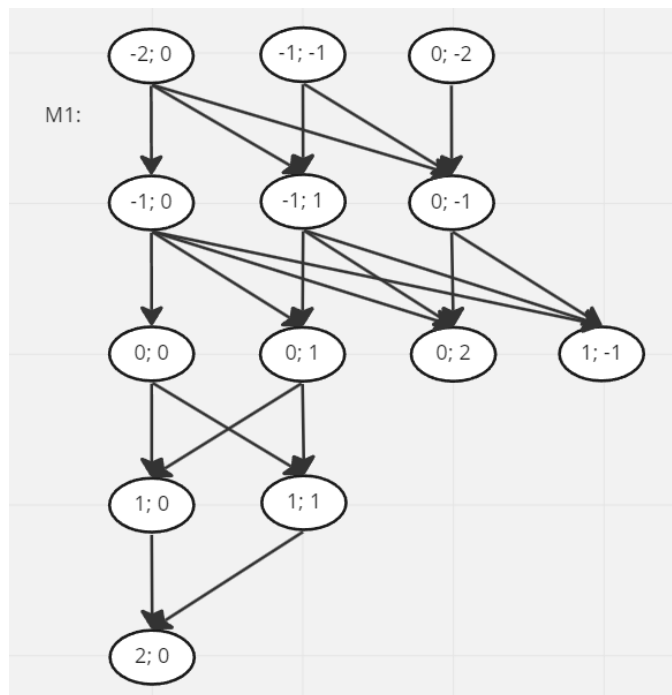
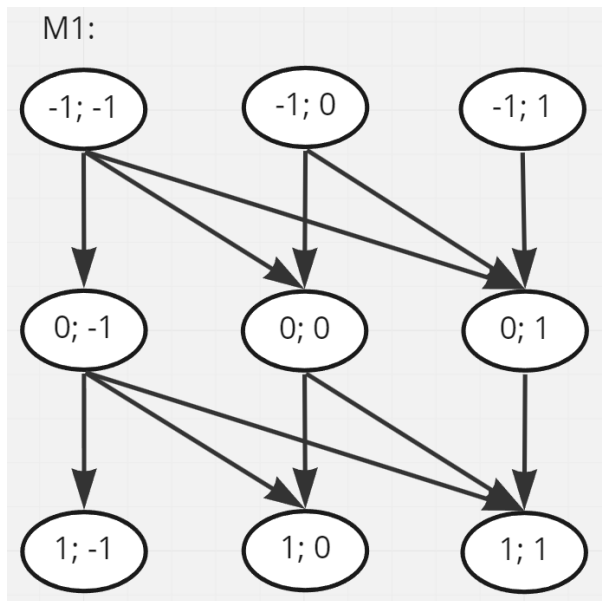
Relation On M2 Topological Sort:

```

(-2; 0) (-1; -1) (0; -2)
(-1; 0) (-1; 1) (0; -1)
(0; 0) (0; 1) (0; 2) (1; -1)
(1; 0) (1; 1)
(2; 0)

```

4. Изобразить диаграмму Хассе отношения доминирования на множествах M1 и M2.



5. Найти минимальные и максимальные элементы множеств M1 и M2.

M1:

Минимальный элемент - $(-1; -1)$ $(-1; 0)$ $(-1; 1)$

Максимальный элемент - $(1; -1)$ $(1; 0)$ $(1; 1)$

M2:

Минимальный элемент - $(-2; 0)$ $(-1; -1)$ $(0; -2)$

Максимальный элемент - $(2; 0)$

6. Найти, если существуют, наименьший и наибольший элементы множеств M1 и M2.

Из диаграммы Хассе:

M1:

Наименьший элемент – отсутствует

Наибольший элемент – отсутствует

M2:

Наименьший элемент – отсутствует

Наибольший элемент – $(2; 0)$

Вывод: в ходе работы были изучены упорядоченные множества, алгоритм топологической сортировки, освоено представление множества диаграммами Хассе, получены навыки нахождения минимальных (максимальных) и наименьших (наибольших) элементов упорядоченного множества