

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)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа №3.4

по дисциплине: Дискретная математика
тема: «Упорядоченные множества»

Выполнил: ст. группы ПВ-212
Степанов Степан Николаевич

Проверили:
Рязанов Юрий Дмитриевич
Бондаренко Татьяна Владимировна

Белгород 2022 г.

Вариант №8

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 M_1 и M_2 .

Пронумеруем точки из множеств M_1 и M_2 :

1	(-1, -1)
2	(-1, 0)
3	(-1, 1)
4	(0, -1)
5	(0, 0)
6	(0, 1)
7	(1, -1)
8	(1, 0)
9	(1, 1)

1	(-2, 0)
2	(-1, -1)
3	(-1, 0)
4	(-1, 1)
5	(0, -2)
6	(0, -1)
7	(0, 0)
8	(0, 1)
9	(0, 2)
10	(1, -1)
11	(1, 0)
12	(1, 1)
13	(2, 0)

```
binaryRelation
getRelationFromPoints(const std::vector<std::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[a].first +
                points[a].second * points[a].second <
                points[b].first * points[b].first +
                points[b].second * points[b].second) {
                res[a][b] = true;
            }
        }
    }

    return res;
}
```

```

int main() {
    std::vector<std::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}};

    std::vector<std::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);

    return 0;
}

```

Получим матрицы отношений на множествах M_1 и M_2 :

0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	1	
0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	1	
1	1	1	1	0	1	1	1	1	
1	0	1	0	0	0	1	0	1	
0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	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	1	0	0	0	1	0	0	0	1
1	1	0	1	1	0	0	0	1	1	0	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	1	1	0	1	1
1	1	1	1	1	1	0	1	1	1	1	1	1
1	1	0	1	1	0	0	0	1	1	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1
1	1	0	1	1	0	0	0	1	1	0	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1
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;
}

```

Получим матрицы отношений доминирования на множествах M_1 и M_2 :

0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	1	
0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	1	
0	1	0	1	0	1	0	1	0	
1	0	1	0	0	0	1	0	1	
0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	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
1	0	0	0	1	0	0	0	1	0	0	0	1	
0	1	0	1	0	0	0	0	0	1	0	1	0	
1	0	0	0	1	0	0	0	1	0	0	0	1	
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	1	0	
0	0	1	0	0	1	0	1	0	0	1	0	0	
0	1	0	1	0	0	0	0	0	1	0	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	
0	1	0	1	0	0	0	0	0	1	0	1	0	
1	0	0	0	1	0	0	0	1	0	0	0	1	
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.

```

std::vector<std::vector<int>>
binaryRelation_topologicalSort(const binaryRelation &relation) {
    binaryRelation dominanceRelation =
        binaryRelation_getDominanceRelation(relation);

    auto dimension = relation.size();

    std::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;
}

int main() {
    std::vector<std::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}};

    std::vector<std::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 rel1 = getRelationFromPoints(M1);
    auto rel2 = getRelationFromPoints(M2);

    auto M1_levels = binaryRelation_topologicalSort(M1, rel1);
    auto M2_levels = binaryRelation_topologicalSort(M2, rel2);

    return 0;
}

```

Получим распределение элементов по уровням (от верхнего к нижнему) на множествах M_1 и M_2 :

```

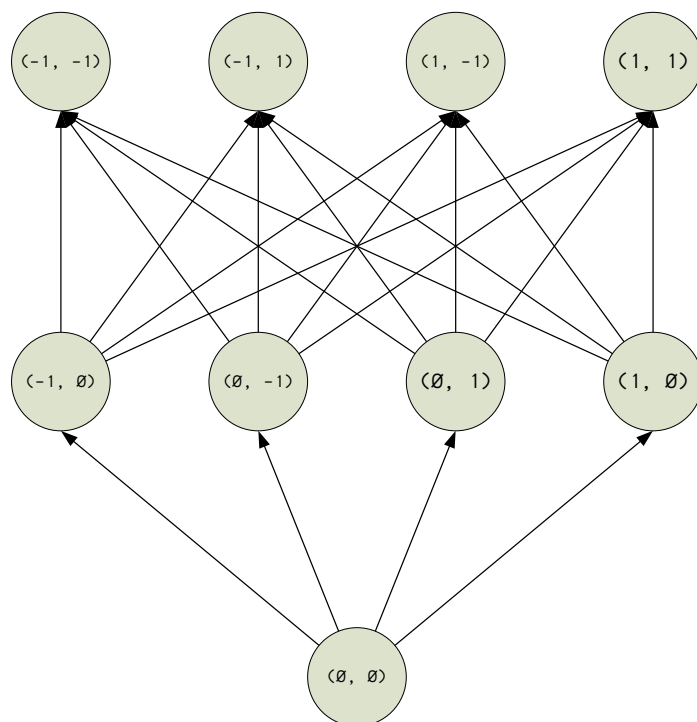
M1 levels:
{ (-1, -1) (-1, 1) (1, -1) (1, 1) }
{ (-1, 0) (0, -1) (0, 1) (1, 0) }
{ (0, 0) }

M2 levels:
{ (-2, 0) (0, -2) (0, 2) (2, 0) }
{ (-1, -1) (-1, 1) (1, -1) (1, 1) }
{ (-1, 0) (0, -1) (0, 1) (1, 0) }
{ (0, 0) }

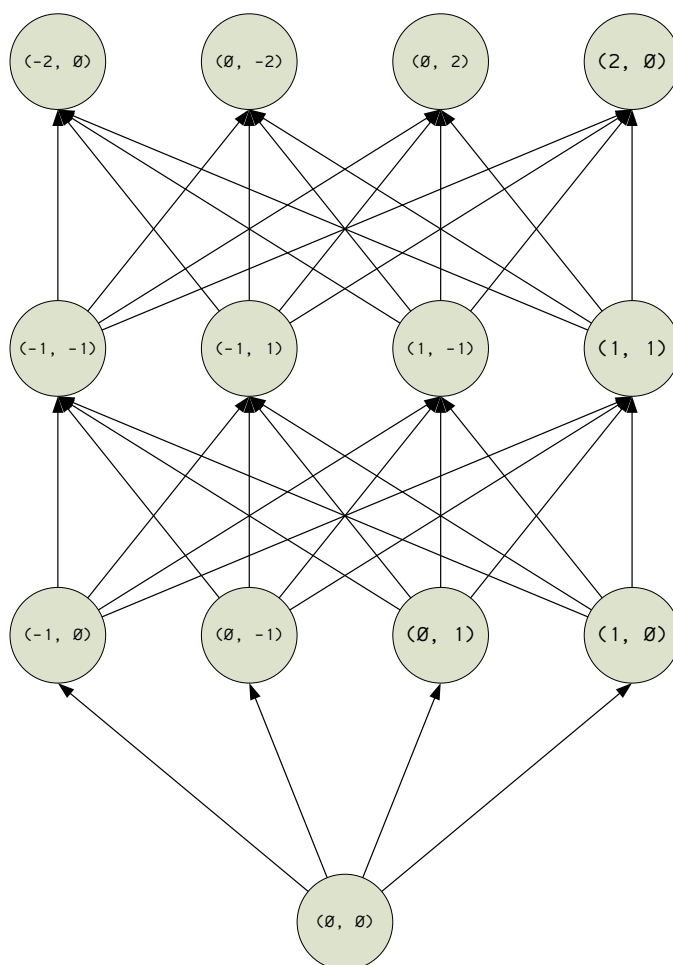
```

4. Изобразить диаграмму Хассе отношения доминирования на множествах M_1 и M_2 .

M_1 :



M_2 :



5. Найти минимальные и максимальные элементы множеств M_1 и M_2 .

Из диаграммы Хассе:

M_1 :

Минимальный элемент – $(0, 0)$

Максимальный элемент – $(-1, -1), (-1, 1), (1, -1), (1, 1)$

M_2 :

Минимальный элемент – $(0, 0)$

Максимальный элемент – $(-2, 0), (0, -2), (0, 2), (2, 0)$

6. Найти, если существуют, наименьший и наибольший элементы множеств M_1 и M_2 .

Из диаграммы Хассе:

M_1 :

Наименьший элемент – $(0, 0)$

Наибольший элемент – отсутствует

M_2 :

Наименьший элемент – $(0, 0)$

Наибольший элемент – отсутствует

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.