**МИНОБРНАУКИ РОССИИ**

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

дисциплина: Дискретная математика

Лабораторная работа №3.4

тема: «Упорядоченные множества»

|  |  |
| --- | --- |
|  | Выполнил: ст. группы ПВ-21  Донцов Александр Алексеевич  Проверил: Рязанов Ю.Д. |

Белгород

2018

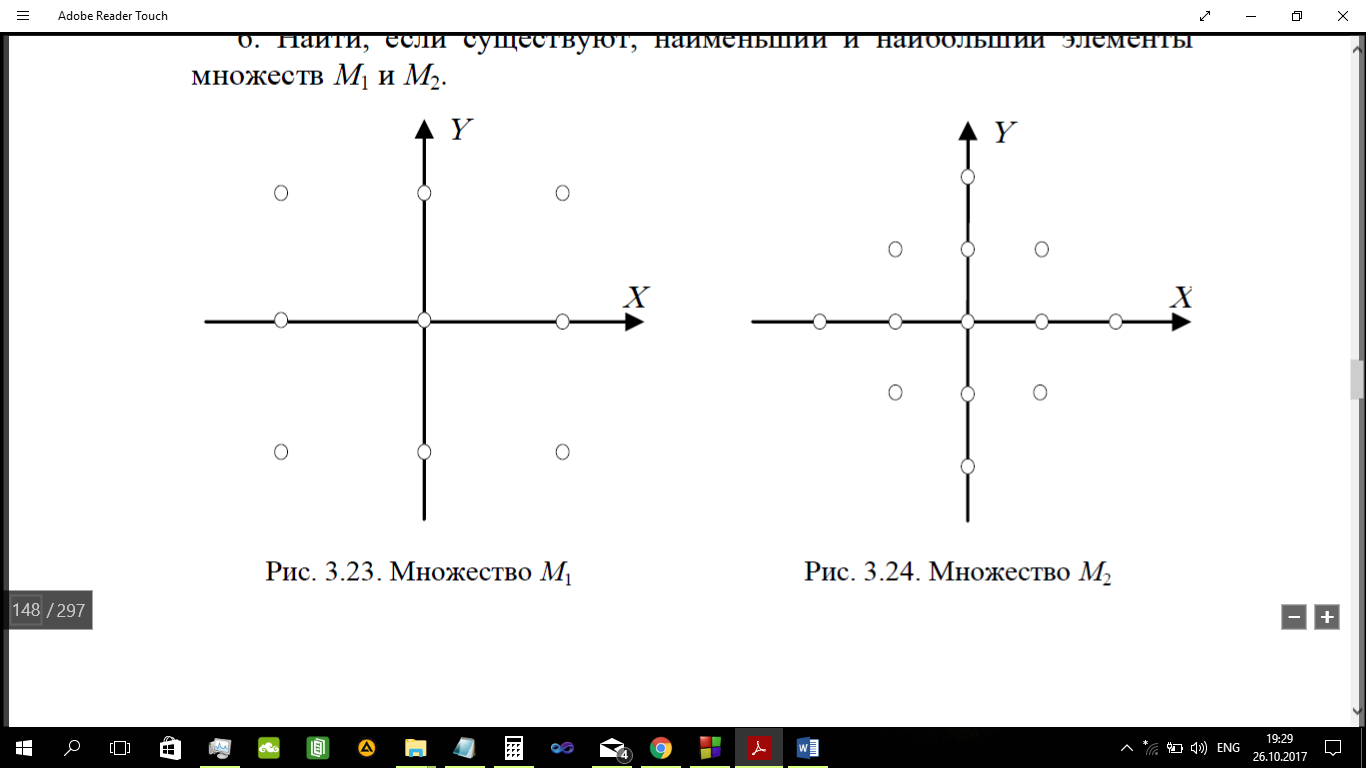
**Цель занятия:** изучить упорядоченные множества, алгоритм топологической сортировки, научиться представлять множества диаграммами Хассе, находить минимальные (максимальные) и наименьшие (наибольшие) элементы упорядоченного множества.

Вариант 1



**Задания**

Даны множества точек на плоскости М1 (рис. 3.23), М2 (рис. 3.24) и отношение порядка (табл. 3.5). Для определения отношения на множестве точек примем следующие обозначения: ax — абсцисса точки a; ay — ордината a. На рис. 3.23 координаты правой верхней точки считать (1,1). На рис. 3.24 координаты самой верхней точки считать (0,2), а координаты самой правой точки считать (2,0).



1. Написать программы, формирующие матрицы отношения порядка, в соответствии с вариантом задания (табл. 3.5), на множествах М1 и М2.

Формирование множеств и матрицы отношений

void formationM1(int rez[N][N]) {

point M1[9];

M1[0].x = -1; M1[5].x = 0;

M1[0].y = 1; M1[5].y = -1;

M1[1].x = -1; M1[6].x = 1;

M1[1].y = 0; M1[6].y = 1;

M1[2].x = -1; M1[7].x = 1;

M1[2].y = 1; M1[7].y = 0;

M1[3].x = 0; M1[8].x = 1;

M1[3].y = 1; M1[8].y = -1;

M1[4].x = 0;

M1[4].y = 0;

clear(rez, 9);

for (int i = 0; i < 9; i++)

for (int j = 0; j < 9; j++)

if (M1[i].x <= M1[j].x && M1[i].y <= M1[j].y)

rez[i][j] = 1;

}

void formationM2(int rez[N][N]) {

point M2[13];

M2[0].x = -2;

M2[0].y = 0;

M2[1].x = -1; M2[7].x = 0;

M2[1].y = 1; M2[7].y = -1;

M2[2].x = -1; M2[8].x = 0;

M2[2].y = 0; M2[8].y = -2;

M2[3].x = -1; M2[9].x = 1;

M2[3].y = 1; M2[9].y = 1;

M2[4].x = 0; M2[10].x = 1;

M2[4].y = 2; M2[10].y = 0;

M2[5].x = 0; M2[11].x = 1;

M2[5].y = 1; M2[11].y = -1;

M2[6].x = 0; M2[12].x = 2;

M2[6].y = 0; M2[12].y = 0;

clear(rez, 13);

for (int i = 0; i < 13; i++)

for (int j = 0; j < 13; j++)

if (M2[i].x <= M2[j].x && M2[i].y <= M2[j].y)

rez[i][j] = 1;

}

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

void dominance(int a[N][N], int n) {

for(int x = 0; x < n; x++)

for(int y = 0; y < n; y++)

for(int z = x + 1; z < y; z++)

if(a[x][z] && a[z][y])

a[x][y] = 0;

}

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

void topsort(int \*\*m, int \*w, int n) {

int x, y, lvl = 0;

for (x = 0; x < n; x++){

w[x] = 0;

for (y = 0; y < n; y++)

w[x] += m[x][y];

}

while (empty(w, n)) {

printf("\nУровень %d: { ", lvl);

lvl++;

for (x = 0; x < n; x++)

if (w[x] == 0){

printf("%i ", x + 1);

w[x] = -1;

}

printf("}\n");

for (x = 0; x < n; x++)

if (w[x] == lvl)

w[x] = 0;

}

}

int empty(int \*w, int n) {

int i = 0;

while (i < n && w[i] < 0)

i++;

return (i < n);

}

4. Изобразить диаграмму Хассе отношения доминирования на множествах М1 и М2.

5. Найти минимальные и максимальные элементы множеств М1 и М2.

|  |  |  |
| --- | --- | --- |
|  | Максимальные элементы | Минимальныe элементы |
| М1 |  |  |
| М2 |  |  |

6. Найти, если существуют, наименьший и наибольший элементы множеств М1 и М2.

|  |  |  |
| --- | --- | --- |
|  | Наибольший элемент | Наименьший элемент |
| М1 |  |  |
| М2 |  |  |

