МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

Кафедра И9 Системы управления и компьютерных технологий
шифр наименование кафедры, по которой выполняется работа

Дисциплина **Представление знаний в информационных системах**
наименование дисциплины

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

3
номер (при наличии)

Разработка программной реализации алгоритма минимаксной процедуры поиска на игровом дереве

при наличии указать тему лабораторной работы и (или) номер варианта

ОБУЧАЮЩИЙСЯ

группы О729Б

подпись Веремчук А.О.
фамилия и инициалы

дата сдачи

ПРОВЕРИЛ

подпись Мариев И.В.
фамилия и инициалы

Оценка / балльная оценка

дата проверки

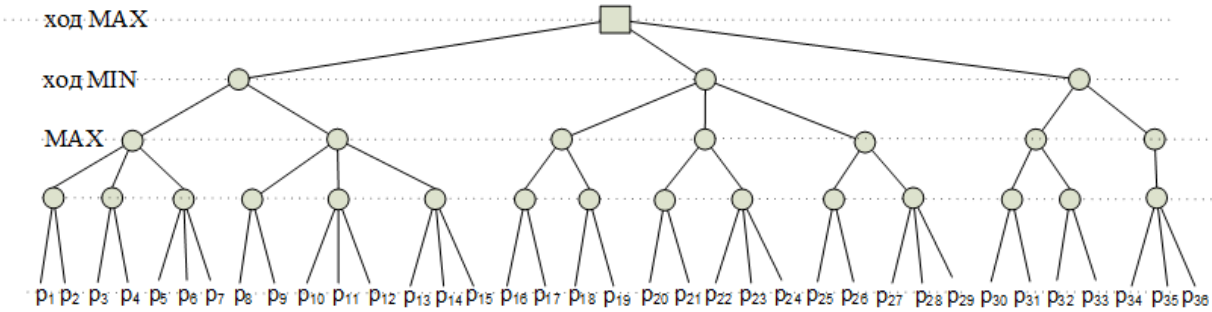
г. Санкт-Петербург
2024 г.

СОДЕРЖАНИЕ

1 Постановка задачи

1. Базовое задание.

Требуется разработать программу, реализующую минимаксный алгоритм выбора наилучшего первого хода на основе анализа возвращенных оценок. Исходные данные согласно индивидуальному варианту задания представлены на рисунке 1 в виде фрагмента игрового дерева с заданными значениями оценок его листьев.



i=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Pi=	0	0	2	1	-2	1	-3	5	4	6	5	5	4	3	7	5	4	2	1	2	1	0	1	2	0	2	-2	1	0	2	-1	0	1	2	3	2

Рисунок 1 – Заданное дерево

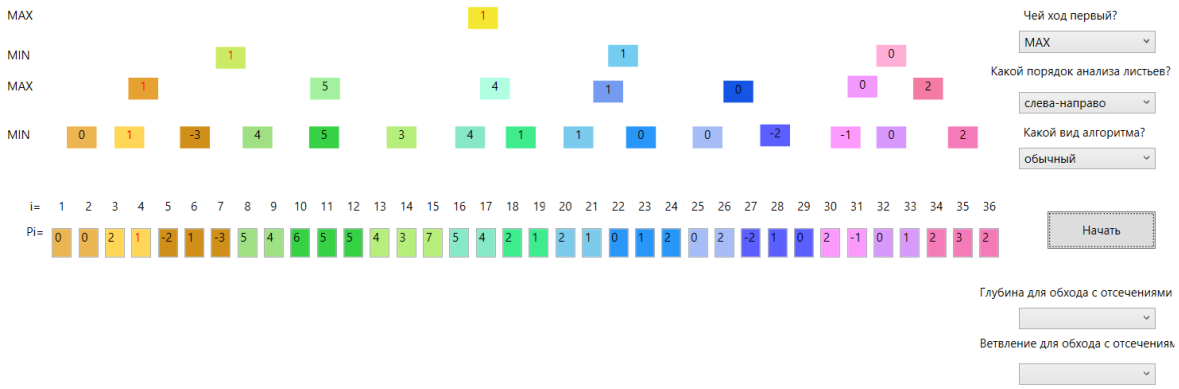
2. Дополнительное задание.

Требуется разработать программу, реализующую минимаксный алгоритм с альфа-бета отсечениями выбора наилучшего первого хода на основе анализа возвращенных оценок. Фрагмент игрового дерева требуется самостоятельно сгенерировать исходя из задаваемых параметров глубины анализа дерева и его ветвления (то есть количества потомков у разных вершин, например, в диапазоне от 1 до 4) и произвести оценку с помощью эвристической функции (случайными значениями) его листьев.

3. Сформировать отчет о выполненных заданиях.

2 Демонстрация работы программы

Интерфейс программы позволяет задавать направление обхода и игрока, который делает первый ход, что представлено на рисунке 2. Так же есть возможность изменять значения листьев по желанию пользователя. Потомки одного родителя выделены одной цветовой гаммой.



После прохождения алгоритма по дереву, красным цветом текста отмечается последовательность позиций в игре, определяющая для обоих игроков оптимальную игру в соответствии с принципами алгоритма минимакс, что представлено на рисунке 3. Так же слева указывается на основе выбора пользователя, чей ход был первым.

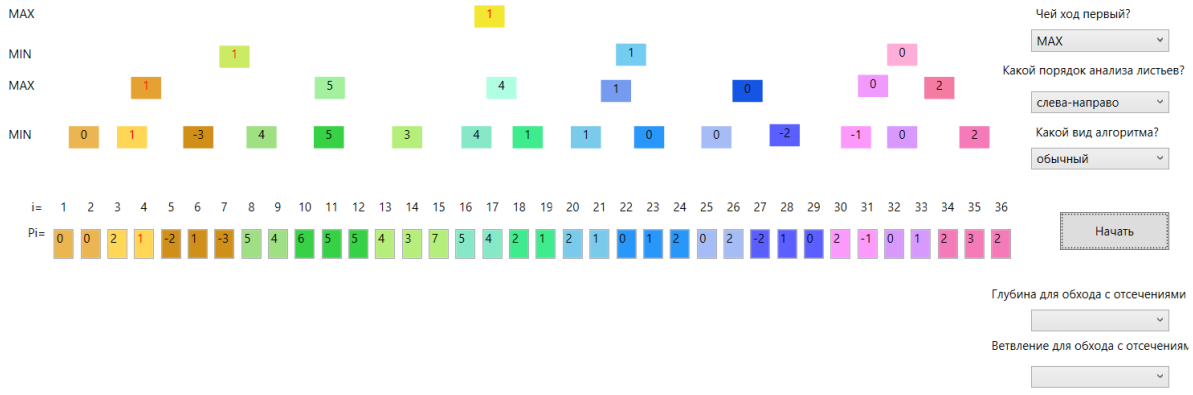


Рисунок 3 – Выделение позиций

На рисунке 4 представлен вариант обхода дерева справа-налево.

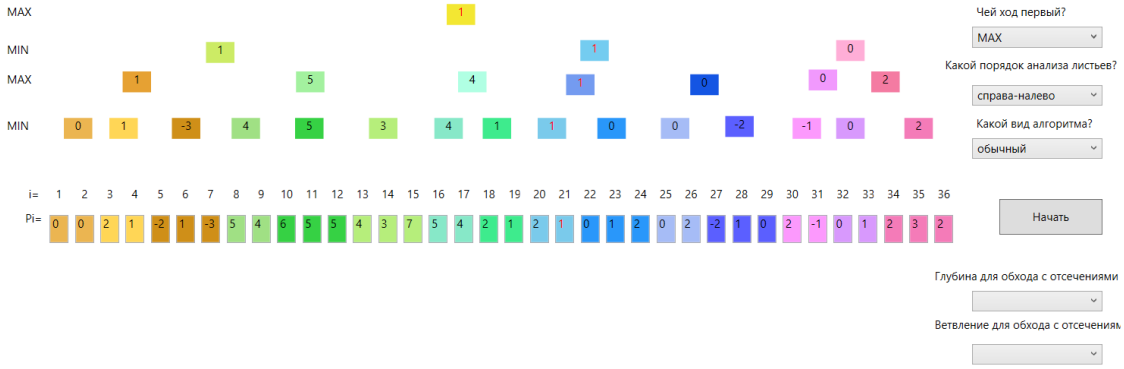


Рисунок 4 – Обход справа-налево

При обходе этого дерева слева-направо путь будет другим, что показано на рисунке 5.

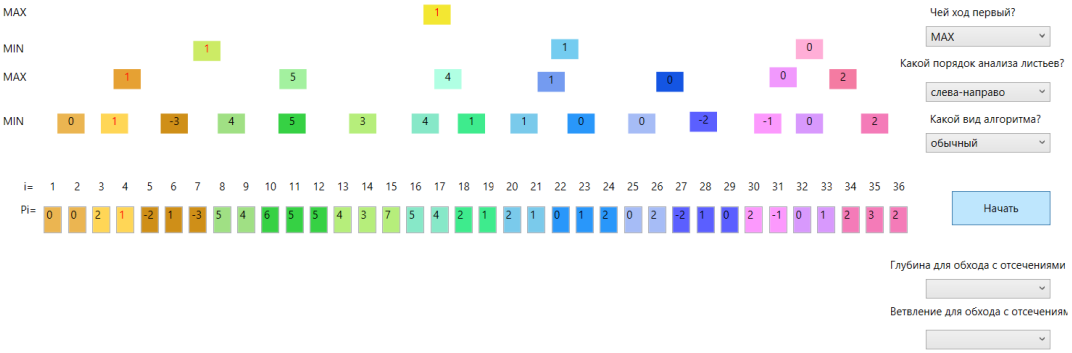


Рисунок 5 – Обход слева-направо

При выборе обхода с отсечениями можно задавать глубину дерева от 2 до 4 и ветвление от 2 до 4, а также настраивать в какую сторону обходить дерево и чей ход будет первый, что показано на рисунке 6. Путь оптимальной игры будет выделен темно-синем, отсекаемые ветви голубым, а цифра, которая записана в узел благодаря отсечению красным.

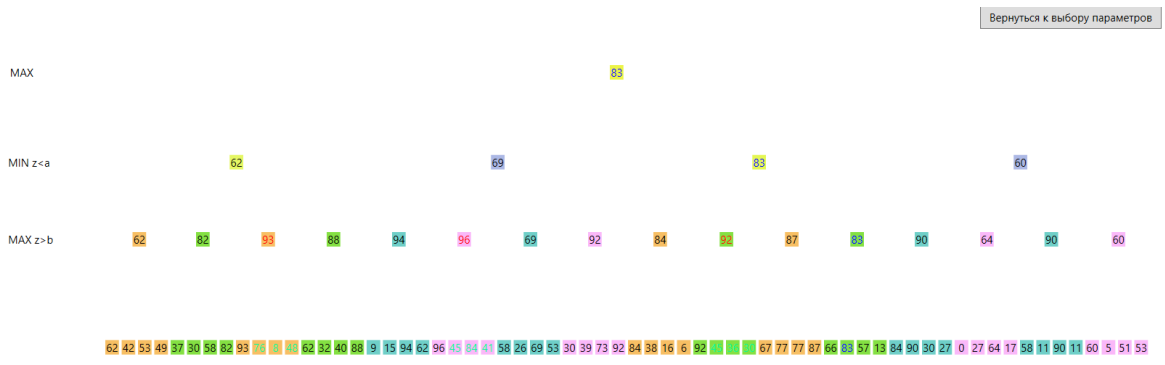


Рисунок 6 – Обход с отсечениями

При обходе справа-налево отсечения тоже меняются, что изображено на рисунке 7.

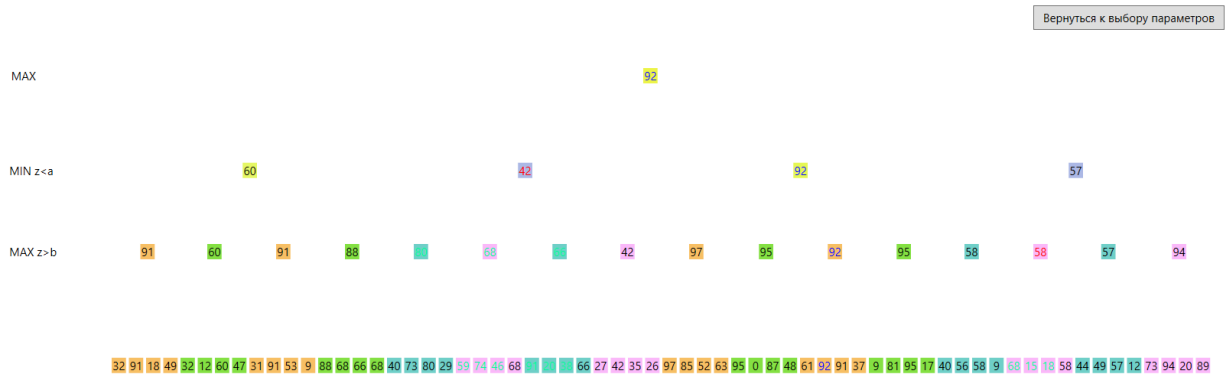


Рисунок 7 – Обход справа-налево

При изменении количества уровней или ветвления дерево видоизменится, что изображено на рисунке 8.

Рисунок 8 – Изменение параметров дерева

3 Листинг кода

Файл “MainWindow.xaml.cs”:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Media.TextFormatting;
using System.Windows.Navigation;
using System.Windows.Shapes;
namespace _3_laba
{
    public partial class MainWindow : Window
    {
        public int[] Lvl_1;
        public int[] Lvl_2;
        public int[] Lvl_3;
        public int[] Lvl_4;
        public int[] Lvl_5;
        public bool f;
        public int imin, imax;
        public MainWindow()
        {
            InitializeComponent();
        }

        private double GetY(TextBlock textBlock)
        {
            textBlock.Measure(new          Size(double.PositiveInfinity,
double.PositiveInfinity));
            textBlock.Arrange(new              Rect(0,               0,
textBlock.DesiredSize.Width, textBlock.DesiredSize.Height));
            var                         position                         =
textBlock.TransformToAncestor(MyGrid).Transform(new Point(0, 0));
            double centerY = position.Y + textBlock.ActualHeight / 2;
            return centerY;
        }
        private double GetX(TextBlock textBlock)
        {
            textBlock.Measure(new          Size(double.PositiveInfinity,
double.PositiveInfinity));
            textBlock.Arrange(new              Rect(0,               0,
textBlock.DesiredSize.Width, textBlock.DesiredSize.Height));
            var                         position                         =
textBlock.TransformToAncestor(MyGrid).Transform(new Point(0, 0));
            double centerX = position.X + textBlock.ActualWidth / 2;
            return centerX;
        }

       public void CreateMas()
        {
            Lvl_5 = new int[36];
            for (int i = 0; i < Lvl_5.Length ; i++){
```

```
                    f = false;
                    TextBox textBox = (TextBox)table.Children[i + 38];
                    string input = textBox.Text;
                    if (string.IsNullOrWhiteSpace(input)){return;}
                    if (!int.TryParse(input, out int number)){return;}
                    Lvl_5[i] = number;
                    f = true;
                }

            }

            private void Button_Click(object sender, RoutedEventArgs e)
            {
                Lvl_1 = new int[1];
                Lvl_2 = new int[3];
                Lvl_3 = new int[7];
                Lvl_4 = new int[15];
                f = false;
                CreateMas();
                if (f == false)
                {
                    MessageBox.Show("Поля должны быть заполнены числами");
                    return;
                }
                else
                {
                    Clear();
                    int a;
                    int.TryParse(vetvlenie.Text, out a);
                    int b;
                    int.TryParse(glubina.Text, out b);
                    if (ask1.Text == "MIN" && ask2.Text == "слева-направо"
&& ask3.Text == "обычный")
                        MinMaxL();
                    else if (ask1.Text == "MIN" && ask2.Text == "справа-
налево" && ask3.Text == "обычный")
                        MinMaxP();
                    else if (ask1.Text == "MAX" && ask2.Text == "слева-
направо" && ask3.Text == "обычный")
                        MaxMinL();
                    else if (ask1.Text == "MAX" && ask2.Text == "справа-
налево" && ask3.Text == "обычный")
                        MaxMinP();
                    else if (ask3.Text == "с отсечениями" && vetvlenie.Text
!= null && glubina.Text != null)
                    {
                        if (a == 0 || b == 0) { MessageBox.Show("Ветвление и
глубина должны быть выбраны"); return; }

                        if (ask1.Text == "MIN" && ask2.Text == "слева-
направо")
                        {
                            var newWindow = new Window1(a, b, 1);
                            newWindow.ShowDialog();
                        }
                        else if (ask1.Text == "MIN" && ask2.Text == "справа-
налево")
                        {
                            var newWindow = new Window1(a, b, 2);
                            newWindow.ShowDialog();
                        }
                        else if (ask1.Text == "MAX" && ask2.Text == "слева-
направо")
```

```
                        {
                            var newWindow = new Window1(a, b, 3);
                            newWindow.ShowDialog();
                        }
                        else if (ask1.Text == "MAX" && ask2.Text == "справа-
налево")
                        {
                            var newWindow = new Window1(a, b, 4);
                            newWindow.ShowDialog();
                        }
                    }
                    return;
                }

            }

            public void Clear()
            {
                for (int i = 0; i < Lvl_5.Length; i++)
                {
                TextBox textBox = (TextBox)table.Children[i + 38];
                textBox.Foreground = Brushes.Black;
                }
                for (int i = 0; i < Lvl_1.Length; i++)
                {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_1_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Black;
                }
                for (int i = 0; i < Lvl_2.Length; i++)
                {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_2_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Black;
                }
                for (int i = 0; i < Lvl_3.Length; i++)
                {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_3_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Black;
                }
                for (int i = 0; i < Lvl_4.Length; i++)
                {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_4_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Black;
                }
            }

            public void MakeMas()
            {
                int a;
                int.TryParse(vetvlenie.Text, out a);
                int b;
                int.TryParse(glubina.Text, out b);

            }

            public void MaxMinL()
            {
                TextBlock_1_0.Text = "MAX";
                TextBlock_2_0.Text = "MIN";
                TextBlock_3_0.Text = "MAX";
```

```
                TextBlock_4_0.Text = "MIN";

                Lvl_4[0] = Math.Min(Lvl_5[0], Lvl_5[1]);
                TextBlock_4_1.Text = (Lvl_4[0]).ToString();
                Lvl_4[1] = Math.Min(Lvl_5[2], Lvl_5[3]);
                TextBlock_4_2.Text = (Lvl_4[1]).ToString();
                Lvl_4[2] = Math.Min(Math.Min(Lvl_5[4], Lvl_5[5]), Lvl_5[6]);
                TextBlock_4_3.Text = (Lvl_4[2]).ToString();
                Lvl_4[3] = Math.Min(Lvl_5[7], Lvl_5[8]);
                TextBlock_4_4.Text = (Lvl_4[3]).ToString();
                Lvl_4[4]    =    Math.Min(Math.Min(Lvl_5[9],    Lvl_5[10]),
Lvl_5[11]);
                TextBlock_4_5.Text = (Lvl_4[4]).ToString();
                Lvl_4[5]    =    Math.Min(Math.Min(Lvl_5[12],    Lvl_5[13]),
Lvl_5[14]);
                TextBlock_4_6.Text = (Lvl_4[5]).ToString();
                Lvl_4[6] = Math.Min(Lvl_5[15], Lvl_5[16]);
                TextBlock_4_7.Text = (Lvl_4[6]).ToString();
                Lvl_4[7] = Math.Min(Lvl_5[17], Lvl_5[18]);
                TextBlock_4_8.Text = (Lvl_4[7]).ToString();
                Lvl_4[8] = Math.Min(Lvl_5[19], Lvl_5[20]);
                TextBlock_4_9.Text = (Lvl_4[8]).ToString();
                Lvl_4[9]    =    Math.Min(Math.Min(Lvl_5[21],    Lvl_5[22]),
Lvl_5[23]);
                TextBlock_4_10.Text = (Lvl_4[9]).ToString();
                Lvl_4[10] = Math.Min(Lvl_5[24], Lvl_5[25]);
                TextBlock_4_11.Text = (Lvl_4[10]).ToString();
                Lvl_4[11]    =    Math.Min(Math.Min(Lvl_5[26],    Lvl_5[27]),
Lvl_5[28]);
                TextBlock_4_12.Text = (Lvl_4[11]).ToString();
                Lvl_4[12] = Math.Min(Lvl_5[29], Lvl_5[30]);
                TextBlock_4_13.Text = (Lvl_4[12]).ToString();
                Lvl_4[13] = Math.Min(Lvl_5[31], Lvl_5[32]);
                TextBlock_4_14.Text = (Lvl_4[13]).ToString();
                Lvl_4[14]    =    Math.Min(Math.Min(Lvl_5[33],    Lvl_5[34]),
Lvl_5[35]);
                TextBlock_4_15.Text = (Lvl_4[14]).ToString();

                Lvl_3[0] = Math.Max(Math.Max(Lvl_4[0], Lvl_4[1]), Lvl_4[2]);
                TextBlock_3_1.Text = (Lvl_3[0]).ToString();
                Lvl_3[1] = Math.Max(Math.Max(Lvl_4[3], Lvl_4[4]), Lvl_4[5]);
                TextBlock_3_2.Text = (Lvl_3[1]).ToString();
                Lvl_3[2] = Math.Max(Lvl_4[6], Lvl_4[7]);
                TextBlock_3_3.Text = (Lvl_3[2]).ToString();
                Lvl_3[3] = Math.Max(Lvl_4[8], Lvl_4[9]);
                TextBlock_3_4.Text = (Lvl_3[3]).ToString();
                Lvl_3[4] = Math.Max(Lvl_4[10], Lvl_4[11]);
                TextBlock_3_5.Text = (Lvl_3[4]).ToString();
                Lvl_3[5] = Math.Max(Lvl_4[12], Lvl_4[13]);
                TextBlock_3_6.Text = (Lvl_3[5]).ToString();
                Lvl_3[6] = Lvl_4[14];
                TextBlock_3_7.Text = (Lvl_3[6]).ToString();


                Lvl_2[0] = Math.Min(Lvl_3[0], Lvl_3[1]);
                TextBlock_2_1.Text = (Lvl_2[0]).ToString();
                Lvl_2[1] = Math.Min(Lvl_3[2], Lvl_3[3]);
                TextBlock_2_2.Text = (Lvl_2[1]).ToString();
                Lvl_2[2] = Math.Min(Lvl_3[4], Lvl_3[5]);
                TextBlock_2_3.Text = (Lvl_2[2]).ToString();

                Lvl_1[0] = Math.Max(Math.Max(Lvl_2[0], Lvl_2[1]), Lvl_2[2]);
                TextBlock_1_1.Text = (Lvl_1[0]).ToString();
```

```
                TextBlock_1_1.Foreground = Brushes.Red;
                int c1 = 0, c2 = 0, c3 = 0;
                for (int i = 0; i < Lvl_2.Length; i++)
                {
                    if (Lvl_2[i] == Lvl_1[0])
                    {
                        c1 = i;
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_2_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        break;
                    }
                }
                int[]a = new int[2];
                a = Chose_c2(c1);
                imin = a[0];
                imax = a[1];
                for (int i = imin; i < imax + 1 ; i++)
                {
                    if (Lvl_3[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_3_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c2 = i;
                        break;
                    }
                }
                a = Chose_c3(c2);
                imin = a[0];
                imax = a[1];
                for (int i = imin; i < imax + 1; i++)
                {
                    if (Lvl_4[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_4_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c3 = i;
                        break ;
                    }
                }
                a = Chose_c4(c3);
                imin = a[0];
                imax = a[1];
                for (int i = imin; i < imax + 1; i++)
                {
                    if (Lvl_5[i] == Lvl_1[0])
                    {
                        TextBox textBox = (TextBox)table.Children[i + 38];
                        textBox.Foreground = Brushes.Red;
                        break;
                    }
                }
            }

             public void MaxMinP()
            {
                TextBlock_1_0.Text = "MAX";
                TextBlock_2_0.Text = "MIN";
                TextBlock_3_0.Text = "MAX";
                TextBlock_4_0.Text = "MIN";
```

```
                Lvl_4[0] = Math.Min(Lvl_5[0], Lvl_5[1]);
                TextBlock_4_1.Text = (Lvl_4[0]).ToString();
                Lvl_4[1] = Math.Min(Lvl_5[2], Lvl_5[3]);
                TextBlock_4_2.Text = (Lvl_4[1]).ToString();
                Lvl_4[2] = Math.Min(Math.Min(Lvl_5[4], Lvl_5[5]), Lvl_5[6]);
                TextBlock_4_3.Text = (Lvl_4[2]).ToString();
                Lvl_4[3] = Math.Min(Lvl_5[7], Lvl_5[8]);
                TextBlock_4_4.Text = (Lvl_4[3]).ToString();
                Lvl_4[4]    =    Math.Min(Math.Min(Lvl_5[9],    Lvl_5[10]),
Lvl_5[11]);
                TextBlock_4_5.Text = (Lvl_4[4]).ToString();
                Lvl_4[5]    =    Math.Min(Math.Min(Lvl_5[12],    Lvl_5[13]),
Lvl_5[14]);
                TextBlock_4_6.Text = (Lvl_4[5]).ToString();
                Lvl_4[6] = Math.Min(Lvl_5[15], Lvl_5[16]);
                TextBlock_4_7.Text = (Lvl_4[6]).ToString();
                Lvl_4[7] = Math.Min(Lvl_5[17], Lvl_5[18]);
                TextBlock_4_8.Text = (Lvl_4[7]).ToString();
                Lvl_4[8] = Math.Min(Lvl_5[19], Lvl_5[20]);
                TextBlock_4_9.Text = (Lvl_4[8]).ToString();
                Lvl_4[9]    =    Math.Min(Math.Min(Lvl_5[21],    Lvl_5[22]),
Lvl_5[23]);
                TextBlock_4_10.Text = (Lvl_4[9]).ToString();
                Lvl_4[10] = Math.Min(Lvl_5[24], Lvl_5[25]);
                TextBlock_4_11.Text = (Lvl_4[10]).ToString();
                Lvl_4[11]    =    Math.Min(Math.Min(Lvl_5[26],    Lvl_5[27]),
Lvl_5[28]);
                TextBlock_4_12.Text = (Lvl_4[11]).ToString();
                Lvl_4[12] = Math.Min(Lvl_5[29], Lvl_5[30]);
                TextBlock_4_13.Text = (Lvl_4[12]).ToString();
                Lvl_4[13] = Math.Min(Lvl_5[31], Lvl_5[32]);
                TextBlock_4_14.Text = (Lvl_4[13]).ToString();
                Lvl_4[14]    =    Math.Min(Math.Min(Lvl_5[33],    Lvl_5[34]),
Lvl_5[35]);
                TextBlock_4_15.Text = (Lvl_4[14]).ToString();


                Lvl_3[0] = Math.Max(Math.Max(Lvl_4[0], Lvl_4[1]), Lvl_4[2]);
                TextBlock_3_1.Text = (Lvl_3[0]).ToString();
                Lvl_3[1] = Math.Max(Math.Max(Lvl_4[3], Lvl_4[4]), Lvl_4[5]);
                TextBlock_3_2.Text = (Lvl_3[1]).ToString();
                Lvl_3[2] = Math.Max(Lvl_4[6], Lvl_4[7]);
                TextBlock_3_3.Text = (Lvl_3[2]).ToString();
                Lvl_3[3] = Math.Max(Lvl_4[8], Lvl_4[9]);
                TextBlock_3_4.Text = (Lvl_3[3]).ToString();
                Lvl_3[4] = Math.Max(Lvl_4[10], Lvl_4[11]);
                TextBlock_3_5.Text = (Lvl_3[4]).ToString();
                Lvl_3[5] = Math.Max(Lvl_4[12], Lvl_4[13]);
                TextBlock_3_6.Text = (Lvl_3[5]).ToString();
                Lvl_3[6] = Lvl_4[14];
                TextBlock_3_7.Text = (Lvl_3[6]).ToString();


                Lvl_2[0] = Math.Min(Lvl_3[0], Lvl_3[1]);
                TextBlock_2_1.Text = (Lvl_2[0]).ToString();
                Lvl_2[1] = Math.Min(Lvl_3[2], Lvl_3[3]);
                TextBlock_2_2.Text = (Lvl_2[1]).ToString();
                Lvl_2[2] = Math.Min(Lvl_3[4], Lvl_3[5]);
                TextBlock_2_3.Text = (Lvl_2[2]).ToString();

                Lvl_1[0] = Math.Max(Math.Max(Lvl_2[0], Lvl_2[1]), Lvl_2[2]);
                TextBlock_1_1.Text = (Lvl_1[0]).ToString();
```

```
                TextBlock_1_1.Foreground = Brushes.Red;
                int c1 = 0, c2 = 0, c3 = 0;
                for (int i = Lvl_2.Length - 1 ; i > 0; i--)
                {
                    if (Lvl_2[i] == Lvl_1[0])
                    {
                        c1 = i;
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_2_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        break;
                    }
                }
                int[]a = new int[2];
                a = Chose_c2(c1);
                imin = a[0];
                imax = a[1];
                for (int i = imax; i > imin - 1 ; i--)
                {
                    if (Lvl_3[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_3_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c2 = i;
                        break;
                    }
                }
                a = Chose_c3(c2);
                imin = a[0];
                imax = a[1];
                for (int i = imax; i > imin - 1; i--)
                {
                    if (Lvl_4[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_4_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c3 = i;
                        break ;
                    }
                }
                a = Chose_c4(c3);
                imin = a[0];
                imax = a[1];
                for (int i = imax; i > imin - 1; i--)
                {
                    if (Lvl_5[i] == Lvl_1[0])
                    {
                        TextBox textBox = (TextBox)table.Children[i + 38];
                        textBox.Foreground = Brushes.Red;
                        break;
                    }
                }
            }

            public void MinMaxL()
            {
                TextBlock_1_0.Text = "MIN";
                TextBlock_2_0.Text = "MAX";
                TextBlock_3_0.Text = "MIN";
                TextBlock_4_0.Text = "MAX";
```

```
                Lvl_4[0] = Math.Max(Lvl_5[0], Lvl_5[1]);
                TextBlock_4_1.Text = (Lvl_4[0]).ToString();
                Lvl_4[1] = Math.Max(Lvl_5[2], Lvl_5[3]);
                TextBlock_4_2.Text = (Lvl_4[1]).ToString();
                Lvl_4[2] = Math.Max(Math.Max(Lvl_5[4], Lvl_5[5]), Lvl_5[6]);
                TextBlock_4_3.Text = (Lvl_4[2]).ToString();
                Lvl_4[3] = Math.Max(Lvl_5[7], Lvl_5[8]);
                TextBlock_4_4.Text = (Lvl_4[3]).ToString();
                Lvl_4[4]    =    Math.Max(Math.Max(Lvl_5[9],    Lvl_5[10]),
Lvl_5[11]);
                TextBlock_4_5.Text = (Lvl_4[4]).ToString();
                Lvl_4[5]    =    Math.Max(Math.Max(Lvl_5[12],    Lvl_5[13]),
Lvl_5[14]);
                TextBlock_4_6.Text = (Lvl_4[5]).ToString();
                Lvl_4[6] = Math.Max(Lvl_5[15], Lvl_5[16]);
                TextBlock_4_7.Text = (Lvl_4[6]).ToString();
                Lvl_4[7] = Math.Max(Lvl_5[17], Lvl_5[18]);
                TextBlock_4_8.Text = (Lvl_4[7]).ToString();
                Lvl_4[8] = Math.Max(Lvl_5[19], Lvl_5[20]);
                TextBlock_4_9.Text = (Lvl_4[8]).ToString();
                Lvl_4[9]    =    Math.Max(Math.Max(Lvl_5[21],    Lvl_5[22]),
Lvl_5[23]);
                TextBlock_4_10.Text = (Lvl_4[9]).ToString();
                Lvl_4[10] = Math.Max(Lvl_5[24], Lvl_5[25]);
                TextBlock_4_11.Text = (Lvl_4[10]).ToString();
                Lvl_4[11]    =    Math.Max(Math.Max(Lvl_5[26],    Lvl_5[27]),
Lvl_5[28]);
                TextBlock_4_12.Text = (Lvl_4[11]).ToString();
                Lvl_4[12] = Math.Max(Lvl_5[29], Lvl_5[30]);
                TextBlock_4_13.Text = (Lvl_4[12]).ToString();
                Lvl_4[13] = Math.Max(Lvl_5[31], Lvl_5[32]);
                TextBlock_4_14.Text = (Lvl_4[13]).ToString();
                Lvl_4[14]    =    Math.Max(Math.Max(Lvl_5[33],    Lvl_5[34]),
Lvl_5[35]);
                TextBlock_4_15.Text = (Lvl_4[14]).ToString();

                Lvl_3[0] = Math.Min(Math.Min(Lvl_4[0], Lvl_4[1]), Lvl_4[2]);
                TextBlock_3_1.Text = (Lvl_3[0]).ToString();
                Lvl_3[1] = Math.Min(Math.Min(Lvl_4[3], Lvl_4[4]), Lvl_4[5]);
                TextBlock_3_2.Text = (Lvl_3[1]).ToString();
                Lvl_3[2] = Math.Min(Lvl_4[6], Lvl_4[7]);
                TextBlock_3_3.Text = (Lvl_3[2]).ToString();
                Lvl_3[3] = Math.Min(Lvl_4[8], Lvl_4[9]);
                TextBlock_3_4.Text = (Lvl_3[3]).ToString();
                Lvl_3[4] = Math.Min(Lvl_4[10], Lvl_4[11]);
                TextBlock_3_5.Text = (Lvl_3[4]).ToString();
                Lvl_3[5] = Math.Min(Lvl_4[12], Lvl_4[13]);
                TextBlock_3_6.Text = (Lvl_3[5]).ToString();
                Lvl_3[6] = Lvl_4[14];
                TextBlock_3_7.Text = (Lvl_3[6]).ToString();

                Lvl_2[0] = Math.Max(Lvl_3[0], Lvl_3[1]);
                TextBlock_2_1.Text = (Lvl_2[0]).ToString();
                Lvl_2[1] = Math.Max(Lvl_3[2], Lvl_3[3]);
                TextBlock_2_2.Text = (Lvl_2[1]).ToString();
                Lvl_2[2] = Math.Max(Lvl_3[4], Lvl_3[5]);
                TextBlock_2_3.Text = (Lvl_2[2]).ToString();

                Lvl_1[0] = Math.Min(Math.Min(Lvl_2[0], Lvl_2[1]), Lvl_2[2]);
                TextBlock_1_1.Text = (Lvl_1[0]).ToString();

                TextBlock_1_1.Foreground = Brushes.Red;
```

```
                int c1 = 0, c2 = 0, c3 = 0;
                for (int i = 0; i < Lvl_2.Length; i++)
                {
                    if (Lvl_2[i] == Lvl_1[0])
                    {
                        c1 = i;
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_2_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        break;
                    }
                }
                int[]a = new int[2];
                a = Chose_c2(c1);
                imin = a[0];
                imax = a[1];
                for (int i = imin; i < imax + 1 ; i++)
                {
                    if (Lvl_3[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_3_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c2 = i;
                        break;
                    }
                }
                a = Chose_c3(c2);
                imin = a[0];
                imax = a[1];
                for (int i = imin; i < imax + 1; i++)
                {
                    if (Lvl_4[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_4_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c3 = i;
                        break ;
                    }
                }
                a = Chose_c4(c3);
                imin = a[0];
                imax = a[1];
                for (int i = imin; i < imax + 1; i++)
                {
                    if (Lvl_5[i] == Lvl_1[0])
                    {
                        TextBox textBox = (TextBox)table.Children[i + 38];
                        textBox.Foreground = Brushes.Red;
                        break;
                    }
                }
            }

            public void MinMaxP()
            {
                TextBlock_1_0.Text = "MIN";
                TextBlock_2_0.Text = "MAX";
                TextBlock_3_0.Text = "MIN";
                TextBlock_4_0.Text = "MAX";

                Lvl_4[0] = Math.Max(Lvl_5[0], Lvl_5[1]);
```

```
                    TextBlock_4_1.Text = (Lvl_4[0]).ToString();
                    Lvl_4[1] = Math.Max(Lvl_5[2], Lvl_5[3]);
                    TextBlock_4_2.Text = (Lvl_4[1]).ToString();
                    Lvl_4[2] = Math.Max(Math.Max(Lvl_5[4], Lvl_5[5]), Lvl_5[6]);
                    TextBlock_4_3.Text = (Lvl_4[2]).ToString();
                    Lvl_4[3] = Math.Max(Lvl_5[7], Lvl_5[8]);
                    TextBlock_4_4.Text = (Lvl_4[3]).ToString();
                    Lvl_4[4]    =    Math.Max(Math.Max(Lvl_5[9],    Lvl_5[10]),
Lvl_5[11]);
                    TextBlock_4_5.Text = (Lvl_4[4]).ToString();
                    Lvl_4[5]    =    Math.Max(Math.Max(Lvl_5[12],    Lvl_5[13]),
Lvl_5[14]);
                    TextBlock_4_6.Text = (Lvl_4[5]).ToString();
                    Lvl_4[6] = Math.Max(Lvl_5[15], Lvl_5[16]);
                    TextBlock_4_7.Text = (Lvl_4[6]).ToString();
                    Lvl_4[7] = Math.Max(Lvl_5[17], Lvl_5[18]);
                    TextBlock_4_8.Text = (Lvl_4[7]).ToString();
                    Lvl_4[8] = Math.Max(Lvl_5[19], Lvl_5[20]);
                    TextBlock_4_9.Text = (Lvl_4[8]).ToString();
                    Lvl_4[9]    =    Math.Max(Math.Max(Lvl_5[21],    Lvl_5[22]),
Lvl_5[23]);
                    TextBlock_4_10.Text = (Lvl_4[9]).ToString();
                    Lvl_4[10] = Math.Max(Lvl_5[24], Lvl_5[25]);
                    TextBlock_4_11.Text = (Lvl_4[10]).ToString();
                    Lvl_4[11]    =    Math.Max(Math.Max(Lvl_5[26],    Lvl_5[27]),
Lvl_5[28]);
                    TextBlock_4_12.Text = (Lvl_4[11]).ToString();
                    Lvl_4[12] = Math.Max(Lvl_5[29], Lvl_5[30]);
                    TextBlock_4_13.Text = (Lvl_4[12]).ToString();
                    Lvl_4[13] = Math.Max(Lvl_5[31], Lvl_5[32]);
                    TextBlock_4_14.Text = (Lvl_4[13]).ToString();
                    Lvl_4[14]    =    Math.Max(Math.Max(Lvl_5[33],    Lvl_5[34]),
Lvl_5[35]);
                    TextBlock_4_15.Text = (Lvl_4[14]).ToString();

                    Lvl_3[0] = Math.Min(Math.Min(Lvl_4[0], Lvl_4[1]), Lvl_4[2]);
                    TextBlock_3_1.Text = (Lvl_3[0]).ToString();
                    Lvl_3[1] = Math.Min(Math.Min(Lvl_4[3], Lvl_4[4]), Lvl_4[5]);
                    TextBlock_3_2.Text = (Lvl_3[1]).ToString();
                    Lvl_3[2] = Math.Min(Lvl_4[6], Lvl_4[7]);
                    TextBlock_3_3.Text = (Lvl_3[2]).ToString();
                    Lvl_3[3] = Math.Min(Lvl_4[8], Lvl_4[9]);
                    TextBlock_3_4.Text = (Lvl_3[3]).ToString();
                    Lvl_3[4] = Math.Min(Lvl_4[10], Lvl_4[11]);
                    TextBlock_3_5.Text = (Lvl_3[4]).ToString();
                    Lvl_3[5] = Math.Min(Lvl_4[12], Lvl_4[13]);
                    TextBlock_3_6.Text = (Lvl_3[5]).ToString();
                    Lvl_3[6] = Lvl_4[14];
                    TextBlock_3_7.Text = (Lvl_3[6]).ToString();

                    Lvl_2[0] = Math.Max(Lvl_3[0], Lvl_3[1]);
                    TextBlock_2_1.Text = (Lvl_2[0]).ToString();
                    Lvl_2[1] = Math.Max(Lvl_3[2], Lvl_3[3]);
                    TextBlock_2_2.Text = (Lvl_2[1]).ToString();
                    Lvl_2[2] = Math.Max(Lvl_3[4], Lvl_3[5]);
                    TextBlock_2_3.Text = (Lvl_2[2]).ToString();

                    Lvl_1[0] = Math.Min(Math.Min(Lvl_2[0], Lvl_2[1]), Lvl_2[2]);
                    TextBlock_1_1.Text = (Lvl_1[0]).ToString();

                    TextBlock_1_1.Foreground = Brushes.Red;
                    int c1 = 0, c2 = 0, c3 = 0;
                    for (int i = Lvl_2.Length - 1 ; i > 0; i--)
```

```
                {
                    if (Lvl_2[i] == Lvl_1[0])
                    {
                        c1 = i;
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_2_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        break;
                    }
                }
                int[]a = new int[2];
                a = Chose_c2(c1);
                imin = a[0];
                imax = a[1];
                for (int i = imax; i > imin - 1 ; i--)
                {
                    if (Lvl_3[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_3_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c2 = i;
                        break;
                    }
                }
                a = Chose_c3(c2);
                imin = a[0];
                imax = a[1];
                for (int i = imax; i > imin - 1; i--)
                {
                    if (Lvl_4[i] == Lvl_1[0])
                    {
                        TextBlock             currentTextBlock             =
(TextBlock)FindName($"TextBlock_4_{i + 1}");
                        currentTextBlock.Foreground = Brushes.Red;
                        c3 = i;
                        break ;
                    }
                }
                a = Chose_c4(c3);
                imin = a[0];
                imax = a[1];
                for (int i = imax; i > imin - 1; i--)
                {
                    if (Lvl_5[i] == Lvl_1[0])
                    {
                        TextBox textBox = (TextBox)table.Children[i + 38];
                        textBox.Foreground = Brushes.Red;
                        break;
                    }
                }
            }

            public int[] Chose_c2(int c1)
            {
                if (c1 == 0) {
                    imin = 0;
                    imax = 1;
                }
                if (c1 == 1) {
                    imin = 2;
                    imax = 3;
                }
```

```
    if (c1 == 2) {
        imin = 4;
        imax = 5;
    }
    int[]a = new int[2];
    a[0] = imin;
    a[1] = imax;
    return a;
}

public int[] Chose_c3(int c2)
{
    if (c2 == 0) {
        imin = 0;
        imax = 2;
    }
    if (c2 == 1) {
        imin = 3;
        imax = 5;
    }
    if (c2 == 2) {
        imin = 6;
        imax = 7;
    }
    if (c2 == 3) {
        imin = 8;
        imax = 9;
    }
    if (c2 == 4)
    {
        imin = 10;
        imax = 11;
    }
    if (c2 == 5) {
        imin = 12;
        imax = 13;
    }
    if (c2 == 6) {
        imin = 14;
        imax = 14;
    }


    int[]a = new int[2];
    a[0] = imin;
    a[1] = imax;
    return a;
}
public int[] Chose_c4(int c3)
{
    if (c3 == 0) {
        imin = 0;
        imax = 1;
    }
    if (c3 == 1) {
        imin = 2;
        imax = 3;
    }
    if (c3 == 2) {
        imin = 4;
        imax = 6;
    }
    if (c3 == 3) {
```

```
                imin = 7;
                imax = 8;
            }
            if (c3 == 4) {
                imin = 9;
                imax = 11;
            }
            if (c3 == 5) {
                imin = 12;
                imax = 14;
            }
            if (c3 == 6) {
                imin = 15;
                imax = 16;
            }
            if (c3 == 7) {
                imin = 17;
                imax = 18;
            }
            if (c3 == 8) {
                imin = 19;
                imax = 20;
            }
            if (c3 == 9) {
                imin = 21;
                imax = 23;
            }
            if (c3 == 10) {
                imin = 24;
                imax = 25;
            }
            if (c3 == 11) {
                imin = 26;
                imax = 28;
            }
            if (c3 == 12) {
                imin = 29;
                imax = 30;
            }
            if (c3 == 13) {
                imin = 31;
                imax = 32;
            }
            if (c3 == 14) {
                imin = 33;
                imax = 35;
            }
            int[]a = new int[2];
            a[0] = imin;
            a[1] = imax;
            return a;
        }
    }
}
```

Файл “Window1.xaml.cs”:

```
   using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace _3_laba
{
    /// <summary>
    /// Логика взаимодействия для Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        public int[] arr;
        public int k = 0;
        public int a;
        public int b;
        public int f;
        public TextBlock currentTextBlock;
        public Window1( int a, int b, int f)
        {
            InitializeComponent();
            int[][] arrays = new int[b][];
            for(int i = 0; i < b ; i++)
            {
                arrays[i] = new int[(int)Math.Pow(a, i)];
                for (int j = 0; j < arrays[i].Length; j++)
                {
                    arrays[i][j] = 0;
                }
            }
            this.a = a;
            this.b = b;
            this.f = f;
            arrays = Sokrutie(arrays);
            if (f == 1)
            { MinMaxL(arrays); FindpathL(arrays); }
            else if (f == 2)
            { MinMaxP(arrays); FindpathP(arrays); }
            else if (f == 3)
            { MaxMinL(arrays); FindpathL(arrays); }
            else if (f == 4)
            { MaxMinP(arrays); FindpathP(arrays); }

        }

        public int[][] Sokrutie(int[][] arrays)
        {
            int x;
            Random random = new Random();
```

```
                for (int i = 0; i < 4; i++) {
                    int c = 0;
                    for (int j = 0; j < Math.Pow(4, i); j++)
                    {
                        if (a == 2) {
                            if (j % 4 == 2 || j % 4 == 3 || c > Math.Pow(a,
i) || (j >= 8 && j <= 15) || j >= 24)
                            {
                                currentTextBlock = (TextBlock)FindName($"t{i
+ 1}_{j + 1}");
                                if (currentTextBlock != null )
                                    currentTextBlock.Visibility            =
Visibility.Hidden;
                            }
                            else
                            {
                                x = random.Next(0, 100);
                                currentTextBlock                           =
(TextBlock)FindName($"t{b}_{j + 1}");
                                if (currentTextBlock != null)
                                {
                                    c++;
                                    if(i == b-1)
                                    {
                                        currentTextBlock.Text              =
x.ToString();
                                        arrays[b-1][k] = x;
                                        k++;
                                    }
                                }
                            }
                        }
                        if (a == 3) {
                            if (j % 4 == 3 || c > Math.Pow(a, i) || (j >= 12
&& j <= 15) || (j >= 28 && j <= 31) || j >= 44)
                            {
                                currentTextBlock = (TextBlock)FindName($"t{i
+ 1}_{j + 1}");
                                if (currentTextBlock != null )
                                    currentTextBlock.Visibility            =
Visibility.Hidden;
                            }
                            else
                            {
                                x = random.Next(0, 100);
                                currentTextBlock                           =
(TextBlock)FindName($"t{b}_{j + 1}");
                                if (currentTextBlock != null)
                                {
                                    c++;
                                    if(i == b-1)
                                    {
                                        currentTextBlock.Text              =
x.ToString();
                                        arrays[b-1][k] = x;
                                        k++;
                                    }
                                }
                            }
                        }
                        if(a == 4)
                        {
                            x = random.Next(0, 100);
```

```
                                    currentTextBlock                                    =
(TextBlock)FindName($"t{b}_{j + 1}");
                                    if (currentTextBlock != null)
                                    {
                                        c++;
                                        if(i == b-1)
                                        {
                                            currentTextBlock.Text                       =
x.ToString();
                                            arrays[b-1][k] = x;
                                            k++;
                                        }
                                    }
                                }
                            }
                        }

                        for (int i = 4; i > b; i--) {
                            for (int j = 0; j < Math.Pow(4, i-1); j++)
                            {
                                currentTextBlock  =  (TextBlock)FindName($"t{i}_{j  +
1}");
                                if (currentTextBlock != null )
                                    currentTextBlock.Visibility = Visibility.Hidden;
                            }
                        }
                        return arrays;
                    }

                    public void FindpathP(int[][] arrays)
                    {
                        int best = arrays[0][0];
                        currentTextBlock = (TextBlock)FindName($"t1_1");
                        currentTextBlock.Foreground = Brushes.Blue;
                        int index1=0, index2=0;
                        for (int i = 1; i < b; i++)
                        {
                            if(a == 4) //a vetvlenie
                            {
                                if (i == 1)
                                {
                                    for (int j = (int)Math.Pow(a, i) - 1; j >=0 ; j-
-)
                                    {
                                        if (arrays[i][j] == best)
                                        {
                                            currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1}");
                                            currentTextBlock.Foreground             =
Brushes.Blue;
                                            index1 = j;
                                            break;
                                        }
                                    }
                                }
                                if (i == 2)
                                {
                                    for (int j = (index1+1) * 4 - 1 ; j >= index1 *
4  ; j--)
                                    {
                                        if (arrays[i][j] == best)
                                        {
```

```
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        index2 = j;
                                        break;
                                    }
                                }
                            }
                        }
                        if (i == 3)
                        {
                            for (int j = (index2+1) * 4 - 1 ; j >= index2 *
4  ; j--)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        break;
                                    }
                                }
                            }
                        }
                    }

                    if(a == 3)
                    {
                        if (i == 1)
                        {
                            for (int j = (int)Math.Pow(a, i) - 1; j >=0 ; j-
-)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1+j/a}");
                                    currentTextBlock.Foreground            =
Brushes.Blue;
                                    index1 = j;
                                    break;
                                }
                            }
                        }
                        if (i == 2)
                        {
                            for (int j = (index1+1) * 3 - 1 ; j >= index1 *
3  ; j--)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1+j/a}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
```

```
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        index2 = j;
                                        break;
                                    }
                                }
                            }
                            if (i == 3)
                            {
                                for (int j = (index2+1) * 3  - 1 ; j >= index2 *
3; j--)
                                {
                                    if (arrays[i][j] == best)
                                    {
                                        if(j/9   ==   0)   currentTextBlock   =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a }");
                                        if(j/9    ==1)    currentTextBlock    =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 4}");
                                        if(j/9    ==2)    currentTextBlock    =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 8}");
                                        if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                        {
                                            currentTextBlock.Foreground        =
Brushes.Blue;
                                            break;
                                        }
                                    }
                                }
                            }
                        }
                        if(a == 2)
                        {
                            if (i == 1)
                            {
                                for (int j =(int) Math.Pow(a, i) - 1; j >=0; j--
)
                                {
                                    if (arrays[i][j] == best)
                                    {
                                        currentTextBlock                       =
(TextBlock)FindName($"t{i+1}_{j + 1+2*(j/a)}");
                                        currentTextBlock.Foreground            =
Brushes.Blue;
                                        index1 = j;
                                        break;
                                    }
                                }
                            }
                            if (i == 2)
                            {
                                for (int j = (index1+1) * 2 - 1; j >=  index1 *
2 ; j--)
                                {
                                    if (arrays[i][j] == best)
                                    {
                                        currentTextBlock                       =
(TextBlock)FindName($"t{i+1}_{j + 1+2*(j/a)}");
                                        if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                        {
```

```
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        index2 = j;
                                        break;
                                    }
                                }
                            }
                        }
                        if (i == 3)
                        {
                            for (int j =(index2+1) * 2 - 1 ; j >= index2 * 2
; j--)
                            {
                                if (arrays[i][j] == best)
                                {
                                    if(index2%4  ==  0)  currentTextBlock  =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a }");
                                    else if(index2%4 ==1) currentTextBlock =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 1}");
                                    else if(index2%4 ==2) currentTextBlock =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 10}");
                                    else if(index2%4 ==3) currentTextBlock =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 11}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        break;
                                    }
                                }
                            }
                        }
                    }
                }
            }

            public void FindpathL(int[][] arrays)
            {
                int best = arrays[0][0];
                currentTextBlock = (TextBlock)FindName($"t1_1");
                currentTextBlock.Foreground = Brushes.Blue;
                int index1=0, index2=0;
                for (int i = 1; i < b; i++)
                {
                    if(a == 4) //a vetvlenie
                    {
                        if (i == 1)
                        {
                            for (int j = 0; j < Math.Pow(a, i); j++)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                       =
(TextBlock)FindName($"t{i+1}_{j + 1}");
                                    currentTextBlock.Foreground            =
Brushes.Blue;
                                    index1 = j;
                                    break;
                                }
                            }
                        }
                        if (i == 2)
```

```
                        {
                            for (int j = index1 * 4; j < (index1+1) * 4 ;
j++)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        index2 = j;
                                        break;
                                    }
                                }
                            }
                        }
                        if (i == 3)
                        {
                            for (int j = index2 * 4; j < (index2+1) * 4 ;
j++)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        break;
                                    }
                                }
                            }
                        }
                    }

                    if(a == 3)
                    {
                        if (i == 1)
                        {
                            for (int j = 0; j < Math.Pow(a, i); j++)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1+j/a}");
                                    currentTextBlock.Foreground            =
Brushes.Blue;
                                    index1 = j;
                                    break;
                                }
                            }
                        }
                        if (i == 2)
                        {
                            for (int j = index1 * 3; j < (index1+1) * 3 ;
j++)
                            {
                                if (arrays[i][j] == best)
```

```
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1+j/a}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        index2 = j;
                                        break;
                                    }
                                }
                            }
                        }
                        if (i == 3)
                        {
                            for (int j = index2 * 3; j < (index2+1) * 3 ;
j++)
                            {
                                if (arrays[i][j] == best)
                                {
                                    if(j/9   ==   0)   currentTextBlock   =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a }");
                                    if(j/9    ==1)    currentTextBlock    =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 4}");
                                    if(j/9    ==2)    currentTextBlock    =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 8}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        break;
                                    }
                                }
                            }
                        }
                    }
                    if(a == 2)
                    {
                        if (i == 1)
                        {
                            for (int j = 0; j < Math.Pow(a, i); j++)
                            {
                                if (arrays[i][j] == best)
                                {
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1+2*(j/a)}");
                                    currentTextBlock.Foreground            =
Brushes.Blue;
                                    index1 = j;
                                    break;
                                }
                            }
                        }
                        if (i == 2)
                        {
                            for (int j = index1 * 2; j < (index1+1) * 2;
j++)
                            {
                                if (arrays[i][j] == best)
                                {
```

```
                                    currentTextBlock                        =
(TextBlock)FindName($"t{i+1}_{j + 1+2*(j/a)}");
                                    if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                    {
                                        currentTextBlock.Foreground        =
Brushes.Blue;
                                        index2 = j;
                                        break;
                                    }
                                }
                            }
                            if (i == 3)
                            {
                                for (int j = index2 * 2; j < (index2+1) * 2;
j++)
                                {
                                    if (arrays[i][j] == best)
                                    {
                                        if(index2%4  ==  0)  currentTextBlock  =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a }");
                                        else if(index2%4 ==1) currentTextBlock =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 1}");
                                        else if(index2%4 ==2) currentTextBlock =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 10}");
                                        else if(index2%4 ==3) currentTextBlock =
(TextBlock)FindName($"t{i + 1}_{j + 1 + j / a + 11}");
                                        if    (currentTextBlock.Foreground    !=
Brushes.MediumSpringGreen)
                                        {
                                            currentTextBlock.Foreground        =
Brushes.Blue;
                                            break;
                                        }
                                    }
                                }
                            }
                        }
                    }
                }

                public void MaxMinL(int[][] arrays)
                {
                    _1.Text = "MAX";
                    _2.Text = "MIN z<a";
                    _3.Text = "MAX z>b";
                    if(b == 2)
                    {
                        _2.Visibility = Visibility.Hidden;
                        _3.Visibility = Visibility.Hidden;
                    }
                    if(b == 3)
                    {
                        _3.Visibility = Visibility.Hidden;
                    }
                    for (int k = b-1; k >= 1; k--)
                    {
                        if (k % 2 == 1)
                        {
                            for (int i = 0; i < Math.Pow(a, k - 1); i++)
                            {
                                for (int j = i * a; j < (i + 1) * a; j += a)
```

```
                                {
                                    if (i % a == 0)
                                    {
                                        int y = 0;
                                        for (int n = i * a; n < (i + 1) * a;
n++)
                                        {
                                            y = Math.Max(arrays[k][n], y);
                                        }
                                        arrays[k - 1][i] = y;
                                        if(a==4 || i==0)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                        currentTextBlock.Text   =   arrays[k   -
1][i].ToString();
                                    }
                                    else
                                    {
                                        if (arrays[k - 1][i - 1] < arrays[k][j])
                                        {
                                            arrays[k - 1][i] = arrays[k][j];
                                            if(a==4 || i==0)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                            currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                            currentTextBlock.Foreground         =
Brushes.Red;
                                            for (int n = i * a + 1 ; n < (i + 1)
* a; n++)
                                            {
                                                if (a == 4 )
                                                    currentTextBlock            =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                else if (a == 3)
                                                {
                                                    if(n/9          ==          0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
                                                    if(n/9 ==1) currentTextBlock
= (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                    if(n/9 ==2) currentTextBlock
= (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                }
                                                else if (a == 2)
                                                {
                                                    if(n/4          ==          0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 1}");
                                                    if(n/4 ==1) currentTextBlock
= (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 11}");
                                                }
                                                if (currentTextBlock != null)
```

```
                                                currentTextBlock.Foreground
= Brushes.MediumSpringGreen;
                                            }
                                        }
                                        else
                                        {
                                            int y = 0;
                                            for (int n = i * a; n < (i + 1) * a;
n++)
                                            {
                                                y = Math.Max(arrays[k][n], y);
                                            }
                                            arrays[k - 1][i] = y;
                                            if(a==4 || i==0)
                                                currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                            currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                        }
                                    }
                                }
                            }
                        }
                        else if (k % 2 == 0)
                        {

                            for (int i = 0; i < Math.Pow(a, k - 1); i++)
                            {
                                for (int j = i * a; j < (i + 1) * a; j += a)
                                {
                                    if (i % a == 0)
                                    {
                                        int y = 100;
                                        for (int n = i * a; n < (i + 1) * a;
n++)
                                        {
                                            y = Math.Min(arrays[k][n], y);
                                        }
                                        arrays[k - 1][i] = y;
                                        if(a==4 || i==0)
                                            currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                        currentTextBlock.Text  =  arrays[k  -
1][i].ToString();

                                    }
                                    else
                                    {
                                        if   (arrays[k   -   1][i   -   1]   >
arrays[k][j])
                                        {
```

```
                                                arrays[k - 1][i] = arrays[k][j];
                                                if(a==4 || i==0)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1}");
                                                else if(a==3)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                                else if(a==2)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                                currentTextBlock.Text = arrays[k
- 1][i].ToString();
                                                currentTextBlock.Foreground    =
Brushes.Red;
                                                for (int n = i * a + 1 ; n < (i
+ 1) * a; n++)
                                                {
                                                    if (a == 4)
                                                        currentTextBlock       =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                    else if (a == 3)
                                                    {
                                                        if(n/9       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
                                                        if(n/9               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                        if(n/9               ==2)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                    }
                                                    else if (a == 2)
                                                    {
                                                        if(n/4       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +1}");
                                                        if(n/4               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +11}");
                                                    }
                                                    if    (currentTextBlock    !=
null)

currentTextBlock.Foreground = Brushes.MediumSpringGreen;
                                                }
                                            }
                                            else
                                            {
                                                int y = 100;
                                                for (int n = i * a; n < (i + 1)
* a; n++)
                                                {
                                                    y  =  Math.Min(arrays[k][n],
y);
                                                }
                                                arrays[k - 1][i] = y;
                                                if(a==4 || i==0)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1}");
                                                else if(a==3)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                                else if(a==2)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                                currentTextBlock.Text = arrays[k
- 1][i].ToString();
```

```
                                        }
                                    }
                                }
                            }
                        }
                    }
                }

                public void MinMaxL(int[][] arrays)
                {
                    _1.Text = "MIN";
                    _2.Text = "MAX z>b";
                    _3.Text = "MIN z<a";
                    if(b == 2)
                    {
                        _2.Visibility = Visibility.Hidden;
                        _3.Visibility = Visibility.Hidden;
                    }
                    if(b == 3)
                    {
                        _3.Visibility = Visibility.Hidden;
                    }
                    for (int k = b-1; k >= 1; k--)
                    {
                        if (k % 2 == 0)
                        {
                            for (int i = 0; i < Math.Pow(a, k - 1); i++)
                            {
                                for (int j = i * a; j < (i + 1) * a; j += a)
                                {
                                    if (i % a == 0)
                                    {
                                        int y = 0;
                                        for (int n = i * a; n < (i + 1) * a;
n++)
                                        {
                                            y = Math.Max(arrays[k][n], y);
                                        }
                                        arrays[k - 1][i] = y;
                                        if(a==4 || i==0)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                        currentTextBlock.Text   =   arrays[k   -
1][i].ToString();
                                    }
                                    else
                                    {
                                        if (arrays[k - 1][i - 1] < arrays[k][j])
                                        {
                                            arrays[k - 1][i] = arrays[k][j];
                                            if(a==4 || i==0)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
```

```
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                            currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                            currentTextBlock.Foreground         =
Brushes.Red;
                                            for (int n = i * a + 1 ; n < (i + 1)
* a; n++)
                                                {
                                                    if (a == 4)
                                                        currentTextBlock        =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                    else if (a == 3)
                                                    {
                                                        if(n/9       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
                                                        if(n/9               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                        if(n/9               ==2)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                    }
                                                    else if (a == 2)
                                                    {
                                                        if(n/4       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +1}");
                                                        if(n/4               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +11}");
                                                    }
                                                    if    (currentTextBlock    !=
null)

currentTextBlock.Foreground = Brushes.MediumSpringGreen;
                                                }
                                        }
                                        else
                                        {
                                            int y = 0;
                                            for (int n = i * a; n < (i + 1) * a;
n++)
                                            {
                                                y = Math.Max(arrays[k][n], y);
                                            }
                                            arrays[k - 1][i] = y;
                                            if(a==4 || i==0)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                            currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                        }
                                    }
                                }
                            }
                        }
                        else if (k % 2 == 1)
                        {

                            for (int i = 0; i < Math.Pow(a, k - 1); i++)
```

```
                                {
                                    for (int j = i * a; j < (i + 1) * a; j += a)
                                    {
                                        if (i % a == 0)
                                        {
                                            int y = 100;
                                            for (int n = i * a; n < (i + 1) * a;
n++)
                                            {
                                                y = Math.Min(arrays[k][n], y);
                                            }
                                            arrays[k - 1][i] = y;
                                            if(a==4 || i==0)
                                                currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock               =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                            currentTextBlock.Text  =  arrays[k  -
1][i].ToString();

                                        }
                                        else
                                        {
                                            if   (arrays[k   -   1][i   -   1]   >
arrays[k][j])
                                            {
                                                arrays[k - 1][i] = arrays[k][j];
                                                if(a==4 || i==0)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1}");
                                                else if(a==3)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                                else if(a==2)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                                currentTextBlock.Text = arrays[k
- 1][i].ToString();
                                                currentTextBlock.Foreground    =
Brushes.Red;
                                                for (int n = i * a + 1 ; n < (i
+ 1) * a; n++)
                                                {
                                                    if (a == 4)
                                                        currentTextBlock       =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                    else if (a == 3)
                                                    {
                                                        if(n/9       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
                                                        if(n/9               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                        if(n/9               ==2)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                    }
                                                    else if (a == 2)
                                                    {
                                                        if(n/4       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +1}");
```

```
                                                    if(n/4              ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +11}");
                                                    }
                                                    if    (currentTextBlock    !=
null)

currentTextBlock.Foreground = Brushes.MediumSpringGreen;
                                                }
                                            }
                                            else
                                            {
                                                int y = 100;
                                                for (int n = i * a; n < (i + 1)
* a; n++)
                                                {
                                                    y  =  Math.Min(arrays[k][n],
y);
                                                }
                                                arrays[k - 1][i] = y;
                                                if(a==4 || i==0)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1}");
                                                else if(a==3)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                                else if(a==2)
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                                currentTextBlock.Text = arrays[k
- 1][i].ToString();
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }

                    public void MaxMinP(int[][] arrays)
                    {
                        _1.Text = "MAX";
                        _2.Text = "MIN z<a";
                        _3.Text = "MAX z>b";
                        if(b == 2)
                        {
                            _2.Visibility = Visibility.Hidden;
                            _3.Visibility = Visibility.Hidden;
                        }
                        if(b == 3)
                        {
                            _3.Visibility = Visibility.Hidden;
                        }
                        for (int k = b-1; k >= 1; k--)
                        {
                            if (k % 2 == 1)
                            {
                                for (int i = (int)Math.Pow(a, k - 1) - 1; i >= 0; i-
-)
                                {
                                    for (int j = (i + 1) * a - 1; j >= i * a; j -=
a)
                                    {
                                        if (i % a == a-1)
```

```
                                    {
                                        int y = 0;
                                        for (int n = (i + 1) * a - 1; n >= i * a
; n--)
                                        {
                                            y = Math.Max(arrays[k][n], y);
                                        }
                                        arrays[k - 1][i] = y;
                                        if(a==4 || i==0)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock                    =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                        currentTextBlock.Text   =   arrays[k   -
1][i].ToString();
                                    }
                                    else
                                    {
                                        if(i+1   <   arrays[k   -   1].Length   &&
arrays[k - 1][i + 1] < arrays[k][j])
                                        {
                                            arrays[k - 1][i] = arrays[k][j];
                                            if(a==4 || i==0)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                            currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                            currentTextBlock.Foreground         =
Brushes.Red;
                                            for (int n = (i + 1) * a - 2  ; n >=
i * a; n--)
                                            {
                                                if (a == 4 )
                                                    currentTextBlock            =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                else if (a == 3)
                                                {
                                                    if(n/9          ==          0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
                                                    if(n/9 ==1) currentTextBlock
= (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                    if(n/9 ==2) currentTextBlock
= (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                }
                                                else if (a == 2)
                                                {
                                                    if(n/4          ==          0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 1}");
                                                    if(n/4 ==1) currentTextBlock
= (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 11}");
                                                }
                                                if (currentTextBlock != null)
```

```
                                                currentTextBlock.Foreground
= Brushes.MediumSpringGreen;
                                            }
                                        }
                                        else
                                        {
                                            int y = 0;
                                            for (int n =(i + 1) * a - 1 ; n >= i
* a; n--)
                                            {
                                                y = Math.Max(arrays[k][n], y);
                                            }
                                            arrays[k - 1][i] = y;
                                            if(a==4 || i==0)
                                                currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                            currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                        }
                                    }
                                }
                            }
                        }
                        else if (k % 2 == 0)
                        {

                            for (int i = (int)Math.Pow(a, k - 1) - 1 ; i >=0 ;
i--)
                            {
                                for (int j = (i + 1) * a - 1 ; j >= i * a; j
-= a)
                                {
                                    if (i % a == a-1)
                                    {
                                        int y = 100;
                                        for (int n = (i + 1) * a - 1; n >= i
* a; n--)
                                        {
                                            y = Math.Min(arrays[k][n], y);
                                        }
                                        arrays[k - 1][i] = y;
                                        if(a==4 || i==0)
                                            currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                        currentTextBlock.Text  =  arrays[k  -
1][i].ToString();

                                    }
                                    else
                                    {
```

```
                                        if(i+1 < arrays[k - 1].Length &&
arrays[k - 1][i + 1] > arrays[k][j])
                                        {
                                            arrays[k - 1][i] = arrays[k][j];
                                            if(a==4 || i==0)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                            currentTextBlock.Text = arrays[k
- 1][i].ToString();
                                            currentTextBlock.Foreground    =
Brushes.Red;
                                            for (int n = (i + 1) * a - 2  ;
n >= i * a; n--)
                                            {
                                                if (a == 4)
                                                    currentTextBlock       =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                else if (a == 3)
                                                {
                                                    if(n/9       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
                                                    if(n/9               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                    if(n/9               ==2)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                }
                                                else if (a == 2)
                                                {
                                                    if(n/4       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +1}");
                                                    if(n/4               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +11}");
                                                }
                                                if    (currentTextBlock    !=
null)
currentTextBlock.Foreground = Brushes.MediumSpringGreen;
                                            }
                                        }
                                        else
                                        {
                                            int y = 100;
                                            for (int n = (i + 1) * a - 1 ; n
>= i * a; n--)
                                            {
                                                y  =  Math.Min(arrays[k][n],
y);
                                            }
                                            arrays[k - 1][i] = y;
                                            if(a==4 || i==0)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
```

```
                                                    currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                                currentTextBlock.Text = arrays[k
- 1][i].ToString();
                                            }
                                        }
                                    }
                                }
                            }
                        }

                        public void MinMaxP(int[][] arrays)
                        {
                            _1.Text = "MIN";
                            _2.Text = "MAX z>b";
                            _3.Text = "MIN z<a";
                            if(b == 2)
                            {
                                _2.Visibility = Visibility.Hidden;
                                _3.Visibility = Visibility.Hidden;
                            }
                            if(b == 3)
                            {
                                _3.Visibility = Visibility.Hidden;
                            }
                            for (int k = b-1; k >= 1; k--)
                            {
                                if (k % 2 == 0)
                                {
                                    for (int i = (int)Math.Pow(a, k - 1) - 1 ; i >=0 ;
i--)
                                    {
                                        for (int j = (i + 1) * a - 1; j >= i * a; j -=
a)
                                        {
                                            if (i % a == a - 1)
                                            {
                                                int y = 0;
                                                for (int n = (i + 1) * a - 1 ; n >= i *
a; n--)
                                                {
                                                    y = Math.Max(arrays[k][n], y);
                                                }
                                                arrays[k - 1][i] = y;
                                                if(a==4 || i==0)
                                                    currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1}");
                                                else if(a==3)
                                                    currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                                else if(a==2)
                                                    currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                                currentTextBlock.Text   =   arrays[k   -
1][i].ToString();
                                            }
                                            else
                                            {
                                                if(i+1   <   arrays[k   -   1].Length   &&
arrays[k - 1][i + 1] < arrays[k][j])
                                                {
                                                    arrays[k - 1][i] = arrays[k][j];
```

```
                                        if(a==4 || i==0)
                                            currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                        currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                        currentTextBlock.Foreground         =
Brushes.Red;
                                        for (int n = (i + 1) * a - 2  ; n >=
i * a; n--)
                                            {
                                                if (a == 4)
                                                    currentTextBlock        =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                else if (a == 3)
                                                {
                                                    if(n/9       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
                                                    if(n/9               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                    if(n/9               ==2)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                }
                                                else if (a == 2)
                                                {
                                                    if(n/4       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +1}");
                                                    if(n/4               ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +11}");
                                                }
                                                if    (currentTextBlock    !=
null)

currentTextBlock.Foreground = Brushes.MediumSpringGreen;
                                                }
                                    }
                                    else
                                    {
                                        int y = 0;
                                        for (int n = (i + 1) * a; n >=i * a
; n--)
                                        {
                                            y = Math.Max(arrays[k][n], y);
                                        }
                                        arrays[k - 1][i] = y;
                                        if(a==4 || i==0)
                                            currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock                =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                        currentTextBlock.Text  =  arrays[k  -
1][i].ToString();
                                    }
                                }
```

```
                            }
                        }
                    }
                    else if (k % 2 == 1)
                    {

                        for (int i = (int)Math.Pow(a, k - 1) -1 ; i >=0; i--
)
                            {
                                for (int j = (i + 1) * a - 1; j >=i * a ; j
-= a)
                                {
                                    if (i % a == a-1)
                                    {
                                        int y = 100;
                                        for (int n = (i + 1) * a - 1; n >=i
* a ; n--)
                                        {
                                            y = Math.Min(arrays[k][n], y);
                                        }
                                        arrays[k - 1][i] = y;
                                        if(a==4 || i==0)
                                            currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1}");
                                        else if(a==3)
                                            currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                        else if(a==2)
                                            currentTextBlock              =
(TextBlock)FindName($"t{k}_{i + 1+2 * (i / a)}");
                                        currentTextBlock.Text  =  arrays[k  -
1][i].ToString();

                                    }
                                    else
                                    {
                                    if(i+1   <   arrays[k   -   1].Length   &&
arrays[k - 1][i + 1] > arrays[k][j])
                                        {
                                            arrays[k - 1][i] = arrays[k][j];
                                            if(a==4 || i==0)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock          =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                            currentTextBlock.Text = arrays[k
- 1][i].ToString();
                                            currentTextBlock.Foreground    =
Brushes.Red;
                                            for (int n = (i + 1) * a - 2  ;
n >= i * a; n--)
                                            {
                                                if (a == 4)
                                                    currentTextBlock       =
(TextBlock)FindName($"t{k + 1}_{n + 1}");
                                                else if (a == 3)
                                                {
                                                    if(n/9       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a }");
```

```
                                                    if(n/9              ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 4}");
                                                    if(n/9              ==2)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a + 8}");
                                                }
                                                else if (a == 2)
                                                {
                                                    if(n/4       ==       0)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +1}");
                                                    if(n/4              ==1)
currentTextBlock = (TextBlock)FindName($"t{k + 1}_{n + 1 + n / a +11}");
                                                }
                                                if    (currentTextBlock   !=
null)

currentTextBlock.Foreground = Brushes.MediumSpringGreen;
                                            }
                                        }
                                        else
                                        {
                                            int y = 100;
                                            for (int n = (i + 1) * a - 1; n
>=i * a ; n--)
                                            {
                                                y  =  Math.Min(arrays[k][n],
y);
                                            }
                                            arrays[k - 1][i] = y;
                                            if(a==4 || i==0)
                                                currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1}");
                                            else if(a==3)
                                                currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+i/a}");
                                            else if(a==2)
                                                currentTextBlock           =
(TextBlock)FindName($"t{k}_{i + 1+2*(i/a)}");
                                            currentTextBlock.Text = arrays[k
- 1][i].ToString();
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
                private void Button_Click(object sender, RoutedEventArgs e)
                {
                    Close();
                }
            }
        }
```

7 Вывод

В ходе проделанной работы, были изучены минимаксный алгоритм и альфа-бета отсечения, создана визуализация дерева с обеспечением различных обходов и выбором первого игрока.