

БУ ВО «Сургутский государственный университет»
Политехнический институт
Кафедра автоматизированных систем обработки информации и управления

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
ПО ТЕМЕ «Задача о ходе коня»
ПО ДИСЦИПЛИНЕ «Интеллектуальные системы»

Выполнил: студент группы №606-12,
Речук Дмитрий Максимович
Принял: ст. преподаватель кафедры АСОИУ,
Гавриленко Анна Владимировна

Сургут 2024

Оглавление

Введение.....	3
Основные функции программы.....	4
Класс <i>Form1</i>	5
Класс <i>KnightTourSolver</i>	7
Вывод программы	8
Заключение	9

Введение

Целью данной работы является разработка программы для решения классической задачи о ходе коня с использованием языка программирования C#. Задача заключается в нахождении пути для шахматного коня, который проходит через все клетки доски ровно один раз.

Программа позволяет пользователю задать размеры доски и начальную позицию коня, после чего находит и визуализирует путь, который конь может пройти.

Программа состоит из двух основных классов: *Form1* и *KnightTourSolver*. Класс *Form1* отвечает за графический интерфейс и взаимодействие с пользователем, а класс *KnightTourSolver* реализует алгоритм решения задачи о туре коня.

Основные функции программы

1. Инициализация доски: Программа создаёт шахматную доску с заданными пользователем размерами.
2. Алгоритм поиска пути коня: Программа использует метод поиска, основанный на алгоритме Варнсдорфа, который выбирает следующие ходы коня исходя из того, сколько свободных клеток доступно после каждого возможного хода. Это помогает минимизировать вероятность тупика, так как предпочтение отдаётся клеткам с наименьшим количеством доступных ходов.
3. Проверка корректности ввода: перед началом поиска программа проверяет корректность введённых пользователем данных: размеры доски и координаты начальной точки.

Класс *Form1*

Этот класс является частью Windows Forms приложения и отвечает за отрисовку шахматной доски, отображение меток и управления событиями.

Поля и Константы

CellSize: Размер клетки шахматной доски (50 пикселей).

Конструктор

```
public Form1()
{
    InitializeComponent();
}
```

Инициализирует компоненты формы.

Методы

1. *DrawChessBoard(int width, int height, List<Point> path)*

Очищает панель от предыдущих элементов и рисует новую шахматную доску заданных размеров с указанным путем коня.

2. *AddColumnLabels(int width)*

Добавляет метки для колонок доски.

3. *AddRowLabels(int height)*

Добавляет метки для строк доски.

4. *CreateLabel(int x, int y, string text)*

Создает метку (Label) для отображения координат.

5. *DrawCells(int width, int height, List<Point> path)*

Рисует клетки доски, чередуя цвета, и добавляет номера шагов коня, если он их проходит.

6. *AddStepNumberToCell(Panel cell, int stepNumber)*

Добавляет номер шага в конкретную клетку.

7. *FindKnightTour(int boardWidth, int boardHeight, Point start)*

Находит путь коня на доске, используя класс *KnightTourSolver*.

8. *Button1_Click(object sender, EventArgs e)*

Обработчик события нажатия кнопки, проверяет входные данные и запускает поиск пути.

9. *TryGetBoardSize(out int width, out int height)*

Проверяет и получает размеры доски.

10. *TryGetStartPosition(int width, int height, out Point start)*

Проверяет и получает начальную позицию коня.

Класс *KnightTourSolver*

Этот класс содержит логику решения задачи о ходе коня.

Алгоритм поиска: используется жадный алгоритм, основанный на методе "обратного хода". Конь делает шаги, выбирая из возможных вариантов тот, который оставляет наименьшее количество доступных ходов в будущем. Это позволяет минимизировать вероятность попадания в тупик.

Порядок выполнения:

- Начинается с заданной начальной позиции.
- Рекурсивно исследует возможные ходы, помечая клетки как посещённые.
- Если достигнута конечная клетка, возвращается полный путь.
- Если не осталось допустимых ходов, выполняется "обратный ход".

Поля

_boardWidth и *_boardHeight*: Размеры доски.

_moves: Возможные перемещения коня.

Конструктор

```
public KnightTourSolver(int boardwidth, int boardHeight)
{
    ...
}
```

Инициализирует размеры доски.

Методы

1. *Solve(Point start)*

Основной метод, который находит путь коня, начиная с заданной позиции.

2. *GetValidMoves(int x, int y, bool[,] visited)*

Возвращает все допустимые ходы коня из текущей позиции, которые еще не были посещены.

3. *IsValid(int x, int y, bool[,] visited)*

Проверяет, является ли клетка допустимой для перемещения (находится ли она в пределах доски и не посещена ли).

4. *GetDegree(int x, int y, bool[,] visited)*

Рассчитывает степень клетки, то есть количество доступных ходов из данной позиции.

Вывод программы

После запуска программы пользователю предлагается ввести размеры шахматной доски (ширину и высоту) и начальную позицию коня (координаты). После нажатия кнопки "Решить":

Если решение найдено, отображается шахматная доска с пронумерованными клетками, представляющими последовательность ходов коня, начиная с начальной позиции. Если решение не найдено, выводится сообщение об отсутствии решения.

Заключение

Программа реализует задачу обхода шахматной доски конём, обеспечивая визуальное представление процесса. Использование эвристического подхода позволяет эффективно находить решения для досок различных размеров. Программа предоставляет пользователю гибкость в настройке параметров доски и отображения процесса решения, что делает её полезной как для демонстрации алгоритмов поиска, так и для учебных целей.