|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
|  | | |
| Курсовая работа | | |
| по дисциплине «Структуры данных и алгоритмы» | | |
|  | | |
|  | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМ-71 |
| Студент: | Востриков В.А. |
| Преподаватель: | Тракимус Ю.В. |
|  |  |
|  |  |
|  |  |
|  |  |
|  | | |
| Новосибирск | | |
| 2019 | | |

1. **Цель работы**

Найти кратчайший путь передвижения ладьи по заданному клеточному полю, соединяющий две указанные его клетки.

1. **Анализ задачи**

2.1. Исходные данные задачи: «карта» шахматной доски с информацией о начальной и конечной точек маршрута, о положении «стен» на «карте».

2.2. Результат: в случае успеха координаты движения ладьи, провала – сообщение об отсутствии решения либо ошибки.

2.3. Решение: улучшить алгоритм волновой трассировки (алгоритм Ли) согласно нашей задачи.

Алгоритм предназначен для поиска кратчайшего пути от стартовой ячейки к конечной ячейке, если это возможно.

Работа алгоритма включает в себя три этапа: **инициализацию**, **распространение волны** и **восстановление пути**.

Во время инициализации строится образ множества ячеек обрабатываемого поля, каждой ячейке приписываются атрибуты проходимости/непроходимости, запоминаются стартовая и финишная ячейки.

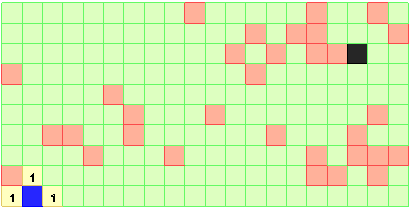
Далее, от стартовой ячейки порождается шаг в соседнюю ячейку, при этом проверяется, проходима ли она, и не принадлежит ли ранее меченной в пути ячейке.

Соседние ячейки принято классифицировать двояко: в смысле окрестности Мура и окрестности фон Неймана, отличающийся тем, что в окрестности фон Неймана соседними ячейками считаются только 4 ячейки по вертикали и горизонтали, в окрестности Мура — все 8 ячеек, включая диагональные (В данном случае классифицируем в смысле окрестности фон Неймана).

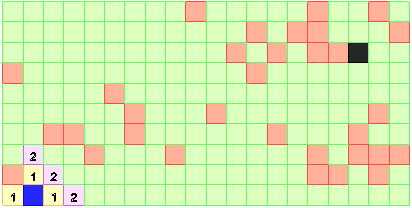
При выполнении условий проходимости и непринадлежности её к ранее помеченным в пути ячейкам, в атрибут ячейки записывается число, равное количеству шагов от стартовой ячейки, от стартовой ячейки на первом шаге это будет 1. Каждая ячейка, меченная числом шагов от стартовой ячейки, становится стартовой и из неё порождаются очередные шаги в соседние ячейки. Очевидно, что при таком переборе будет найден путь от начальной ячейки к конечной, либо очередной шаг из любой порождённой в пути ячейки будет невозможен.

Восстановление кратчайшего пути происходит в обратном направлении: при выборе ячейки от финишной ячейки к стартовой на каждом шаге выбирается ячейка, имеющая атрибут расстояния от стартовой на единицу меньше текущей ячейки. Очевидно, что таким образом находится кратчайший путь между парой заданных ячеек. Трасс с минимальной числовой длиной пути, как при поиске пути в окрестностях Мура, так и фон Неймана может существовать несколько. Выбор окончательного пути в приложениях диктуется другими соображениями, находящимися вне этого алгоритма. Например, при трассировке печатных плат — минимумом линейной длины проложенного проводника.

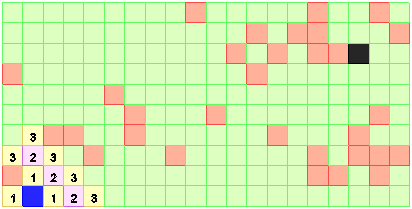
Для лучшего понимания ниже приведены рисунки этапов распространения волны и восстановления пути.



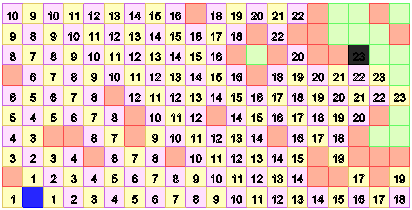
*Рис.1. Этап распространения волны №1.*



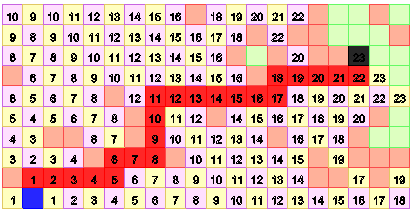
*Рис.2. Этап распространения волны №2.*



*Рис.3. Этап распространения волны №3.*



*Рис.4. Этап распространения волны №4 (конец алгоритма распространения).*



*Рис.5. Этап восстановления пути.*

На картинках выше показана работа алгоритма Ли в классификации окрестности фон Неймана.

В таком случае необходимо понимать, что работа алгоритма Ли не будет учитывать количество совершенных ходов ладьи для достижения конечной клетки.

*Тогда идеальным вариантом для решения этой задачи является решение, которое учитывает наименьшее количество ходов и наименьшее количество пройденных клеток ладьи* (программа, которая будет приведена ниже, решает только задачу с наименьшим количеством ходов).

**Постановка задачи:**

Найти путь с наименьшим количеством ходов ладьи до конечной клетки.

1. **Представление данных**

3.1. Входные данные:

В качестве входных данных из файла считывается двумерный массив объектов (object\_matrix[][]), который содержит информацию о положениях конечной и начальной позиции и стен.

3.2. Выходные данные:

В качестве выходных данных в файл выводятся координаты точек маршрута из массивов