Implementation of Single Layer Maze Router in C++

Md Maksud-Ul-Kabir Rico
Dept. of ECE, UWindosr
Email ID: ricom@uwindsor.ca

Maze Routing:

Maze routing is one kind of global routing. It is similar breadth-first search and it guarantees to find the optimal solution if it exists. The Lee-Moore algorithm is one possible solution for maze routing problems based on breadth-first search. The pseudocode is given below.

```
Algorithm LEE-ROUTER (B, s, t, P)
    input: B, s, t
    output: P
begin
    plist = s;     // "present" list, source of wave exp.
    nlist = φ;     // "next" list (of grid elements)
    temp = 1;      // label number used during wave exp.
    path_exists = FALSE;
    while plist ≠ φ do   // until wave exp. is possible
        for each vertex v_i in plist do
            for each vertex v_j neighboring v_i do
            if B[v_j] = UNBLOCKED then
                L[v_j] = temp;   // label grid element (1, 2, 3, ....)
                INSERT(v_j, nlist);   // add to list for future wave exp.
                if v_j = t then
                    path_exists = TRUE;
                    exit while;
        temp=temp + 1;   // increment label #
        plist = nlist;   // new source(s) for wave exp.
        nlist = φ;
    if path_exists = TRUE then RETRACE (L, P);   → pick one routing path (we may have one or more avail)
    else path does not exist;   // declare failure
end.
```

Figure 1: The pseudocode of Lee-Moore algorithm

Comments on Implementation:

This project was implemented in C++ language using Code::Blocks. Graph, Array, and Queue Data Structures were used to complete the assignment. C++ Standard Template Library (STL), such as vectors and pair were also used. Details are given in the code's comment section.

Result:

Testcase No. 1

The Grid Size : 15

The obstruction cells, source cells, destination cells and the path

obstruction 3 12

obstruction 3 11

obstruction 3 10

obstruction 3 3

obstruction 3 4

obstruction 3 5

obstruction 3 6

obstruction 4 6

obstruction 5 6

obstruction 9 10

obstruction 10 10

obstruction 11 10

obstruction 11 9

obstruction 11 8

obstruction 6 2

obstruction 7 2

obstruction 8 2

obstruction 8 5

obstruction 9 5

obstruction 10 5

obstruction 11 5

obstruction 12 5

The Source Cell: (3, 12)
The Destination Cell: (12, 5)

The Shortest Path:-

(3, 12) -> (4, 12) -> (5, 12) -> (6, 12) -> (7, 12) -> (8, 12) -> (9, 12) -> (10, 12) -> (11, 12) -> (12, 12) -> (12, 11) -> (12, 10) -> (12, 9) -> (12, 8) -> (12, 7) -> (12, 6) -> (12, 5)

The Source Cell: (3, 6)
The Destination Cell: (10, 10)

The Shortest Path:-

The Given Route is not possible

The Source Cell: (8, 5)
The Destination Cell: (8, 2)

The Shortest Path:-

(8, 5) -> (8, 4) -> (8, 3) -> (8, 2)

The Source Cell: (3, 3)
The Destination Cell: (3, 10)

The Shortest Path:-

(3, 3) -> (2, 3) -> (2, 4) -> (2, 5) -> (2, 6) -> (2, 7) -> (2, 8) -> (2, 9) -> (2, 10) -> (3, 10)

Testcase No. 2

The Grid Size: 15

The obstruction cells, source cells, destination cells and the path

obstruction 0 9

obstruction 1 9

obstruction 2 9

obstruction 2 3

obstruction 2 4

obstruction 2 5

obstruction 3 3

obstruction 3 4

obstruction 3 5

obstruction 5 13

obstruction 6 13

obstruction 7 13

obstruction 8 13

obstruction 6 8

obstruction 7 8

obstruction 8 8

obstruction 6 9

obstruction 7 9

obstruction 8 9

obstruction 6 10

obstruction 7 10

obstruction 8 10

obstruction 6 1

obstruction 7 1

obstruction 10 3

obstruction 11 3

obstruction 12 3

obstruction 10 4

obstruction 11 4

obstruction 12 4

obstruction 10 5

obstruction 11 5

obstruction 12 5

obstruction 12 8

obstruction 12 9

obstruction 12 10

obstruction 12 11

obstruction 12 12

The Source Cell: (3, 3)
The Destination Cell: (10, 4)

The Shortest Path:-

The Given Route is not possible

The Source Cell: (2, 9)
The Destination Cell: (10, 5)

The Shortest Path:-

The Given Route is not possible

The Source Cell: (12, 8)
The Destination Cell: (12, 12)

The Shortest Path:-

The Given Route is not possible

The Source Cell: (3, 4)
The Destination Cell: (10, 3)

The Shortest Path:-

The Given Route is not possible

The Source Cell: (7, 1)
The Destination Cell: (7, 13)

The Shortest Path:-

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

The Source Cell: (2, 5)
The Destination Cell: (6, 1)

The Shortest Path:-

(2, 5) -> (1, 5) -> (1, 4) -> (1, 3) -> (1, 2) -> (2, 2) -> (3, 2) -> (4, 2) -> (4, 1) -> (5, 1) -> (6, 1)

Reference:

[1] Dr. Khalid**, “**Routing PDF File from Physical Design Automation for VLSI & FPGAs Course”, Dept. of ECE, University of Windsor.

[2] “Shortest Path in a Binary Maze”, *GeeksforGeeks* [Online], Available: https://www.geeksforgeeks.org/shortest-path-in-a-binary-maze/ [Accessed: April 02, 2020]