# Maze Router

Youssef Mentawy

# Algorithm of Choice

- Lee's Algorithm
  - Objective: Find the optimal path in a 3D grid.
  - Wave Propagation: BFS-based cost computation.
  - Movement Constraints: Layer-specific rules and penalties.
  - Cost Factors: Via transitions and bends.
  - Path Reconstruction: Trace back using parent nodes.
  - Output: Path and total cost.

# Code Structure

- LeeRouter(width, height, bend_penalty, via_penalty)
  - Sets grid dimensions and penalties.
  - Creates two 2D layers (self.layers) to represent the grid.

- add_obstacle(layer, x, y)
  - Marks a grid cell as blocked (-1).
  - Ensures the router avoids these points during pathfinding.

- route_net(net_name, pins)
  - Routes between multiple pins sequentially.
  - Uses _lee_route to calculate sub-paths and aggregate cost.

# Lee's Operation

- Wave Propagation (BFS):

  ➔ _lee_route(start_layer, start_x, start_y, end_layer, end_x, end_y)
  - Explores 6 possible directions (±x, ±y, ±layer).
  - Cost Calculation:
    - ❏ 1: Basic movement cost.
    - ❏ bend_penalty: Applied when direction changes.
    - ❏ via_penalty: Applied when changing layers.

- Constraints:

  ➔ Prevents movement:
  - Into obstacles (-1 in grid).
  - Between invalid layers or out-of-bounds coordinates.
  - Horizontal moves on Layer 1 and vertical moves on Layer 0.

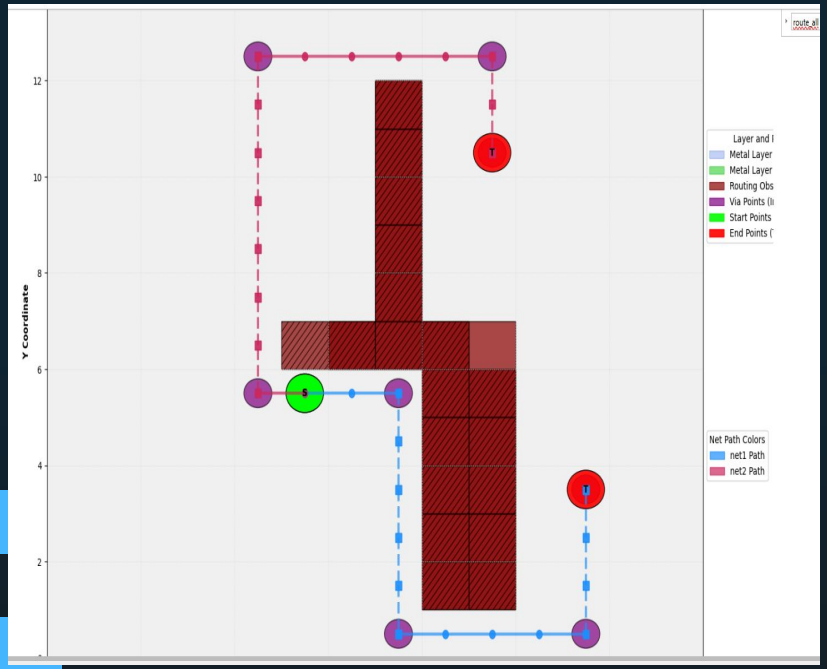
- Path Reconstruction:

  ➔ Backtracks from end to start using came_from.
  ➔ Returns the path and total routing cost.

# Ordering Heuristic

- Heuristic:
  - Prioritize nets with more pins and shorter distances between pins.
  - Sorting Criteria:
    - ❏ Negative number of pins (more pins get higher priority).
    - ❏ Total Manhattan distance (shorter distances are prioritized).
- Outcome:
  - Complex nets are routed first, minimizing congestion and rerouting.

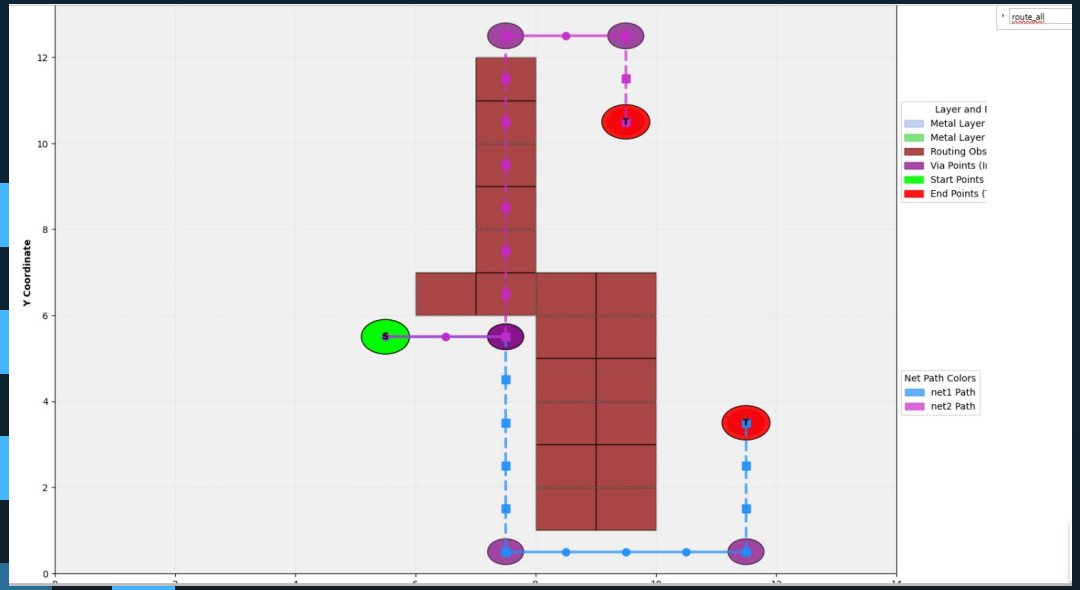
# Test case 1




```
14, 14, 0, 0
OBS (0, 8, 1)
OBS (0, 8, 2)
OBS (0, 8, 3)
OBS (0, 8, 4)
OBS (0, 8, 5)
OBS (0, 8, 6)
OBS (0, 9, 1)
OBS (0, 9, 2)
OBS  (0, 9, 3)
OBS (0, 9, 4)
OBS (0, 9, 5)
OBS (0, 9, 6)
OBS (0, 7, 6)
OBS (0, 6, 6)
OBS  (0, 7, 7)
OBS (0, 7, 8)
OBS (0, 7, 9)
OBS (0, 7, 10)
OBS (0, 7, 11)
OBS (1, 8, 1)
OBS (1, 8, 2)
OBS (1, 8, 3)
OBS (1, 8, 4)
OBS (1, 8, 5)
OBS (1, 8, 6)
OBS (1, 9, 1)
OBS (1, 9, 2)
OBS (1, 9, 3)
OBS (1, 9, 4)
OBS (1, 9, 5)
OBS (1, 7, 6)
OBS (1, 6, 6)
OBS (1, 5, 6)
OBS (1, 7, 7)
OBS (1, 7, 8)
OBS (1, 7, 9)
OBS (1, 7, 10)
OBS (1, 7, 11)


net1 (0, 5, 5) (0, 11, 3)
net2 (0, 5, 5) (0, 9, 10)
```

# Test case 2-removed layer 1 obstacles



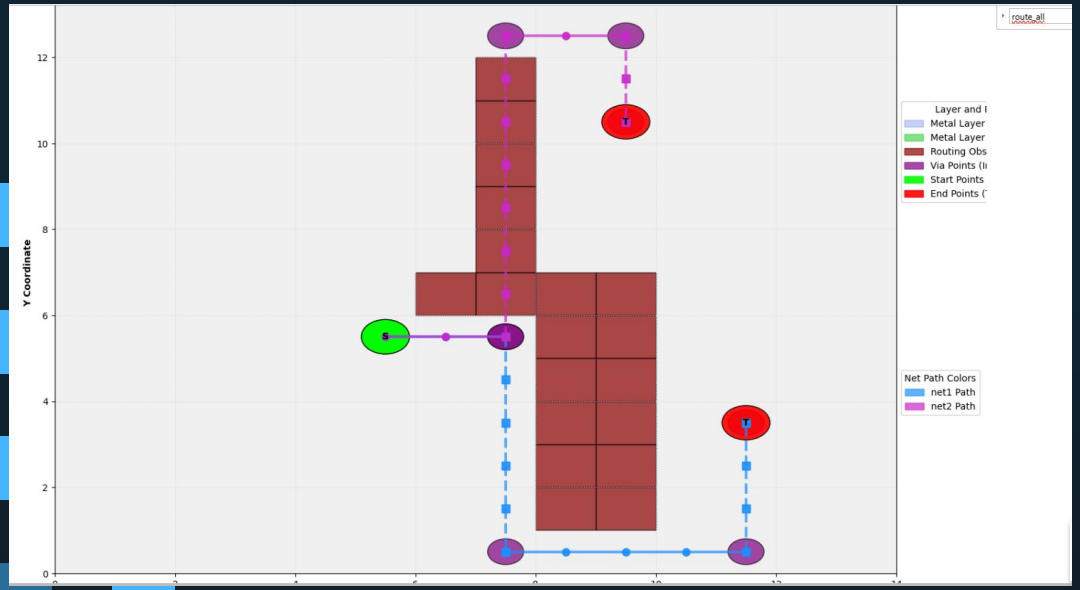

```
14, 14, 0, 0
OBS (0, 8, 1)
OBS (0, 8, 2)
OBS (0, 8, 3)
OBS (0, 8, 4)
OBS (0, 8, 5)
OBS (0, 8, 6)
OBS (0, 9, 1)
OBS (0, 9, 2)
OBS  (0, 9, 3)
OBS (0, 9, 4)
OBS (0, 9, 5)
OBS (0, 9, 6)
OBS (0, 7, 6)
OBS (0, 6, 6)
OBS  (0, 7, 7)
OBS (0, 7, 8)
OBS (0, 7, 9)
OBS (0, 7, 10)
OBS (0, 7, 11)



net1 (0, 5, 5) (0, 11, 3)
net2 (0, 5, 5) (0, 9, 10)
```

net1 routed with cost: 14.00
net2 routed with cost: 15.00

# Test case 2-removed layer 1 obstacles



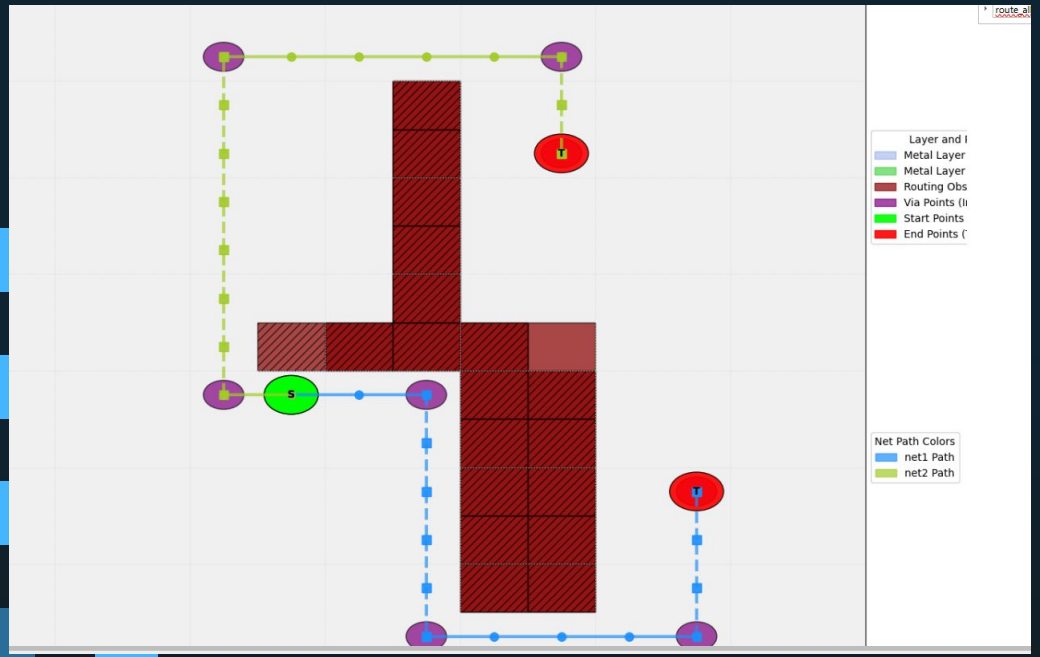

net1 routed with cost: 14.00
net2 routed with cost: 13.00

```
14, 14, 0, 0
OBS (0, 8, 1)
OBS (0, 8, 2)
OBS (0, 8, 3)
OBS (0, 8, 4)
OBS (0, 8, 5)
OBS (0, 8, 6)
OBS (0, 9, 1)
OBS (0, 9, 2)
OBS  (0, 9, 3)
OBS (0, 9, 4)
OBS (0, 9, 5)
OBS (0, 9, 6)
OBS (0, 7, 6)
OBS (0, 6, 6)
OBS  (0, 7, 7)
OBS (0, 7, 8)
OBS (0, 7, 9)
OBS (0, 7, 10)
OBS (0, 7, 11)



net1 (0, 5, 5) (0, 11, 3)
net2 (0, 5, 5) (0, 9, 10)
```

# Test case 3-high bend penalty

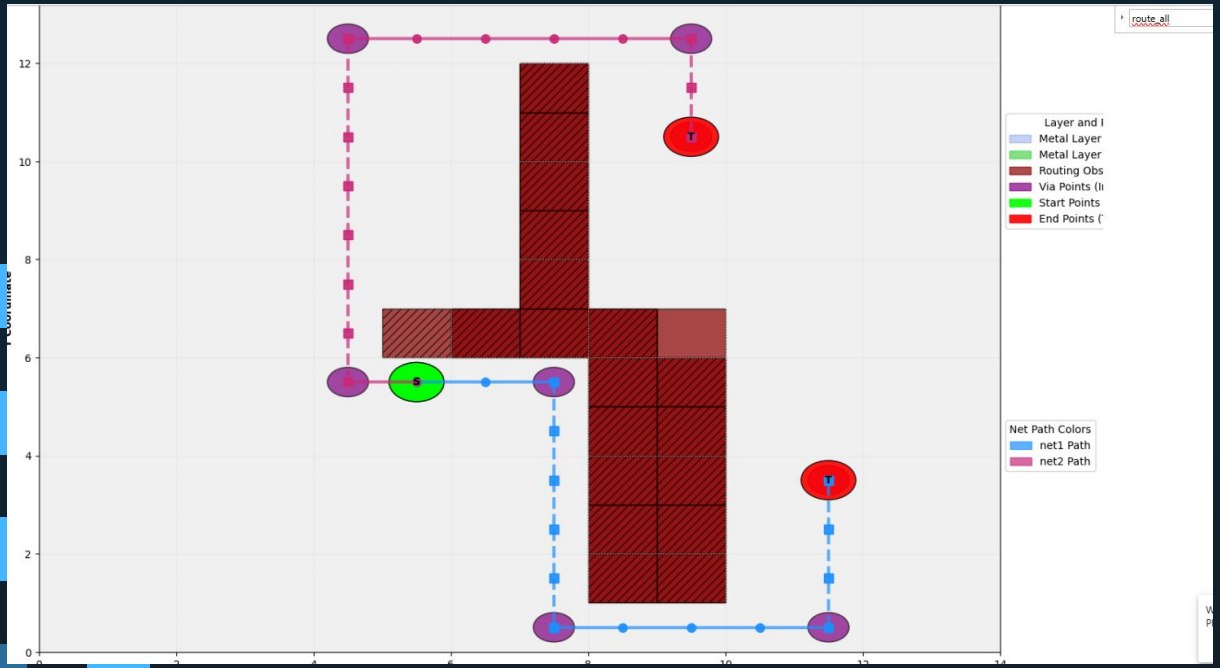



net1 routed with cost: 714.00
net2 routed with cost: 715.00

```
14, 14, 100, 0
OBS (0, 8, 1)
OBS (0, 8, 2)
OBS (0, 8, 3)
OBS (0, 8, 4)
OBS (0, 8, 5)
OBS (0, 8, 6)
OBS (0, 9, 1)
OBS (0, 9, 2)
OBS  (0, 9, 3)
OBS (0, 9, 4)
OBS (0, 9, 5)
OBS (0, 9, 6)
OBS (0, 7, 6)
OBS (0, 6, 6)
OBS  (0, 7, 7)
OBS (0, 7, 8)
OBS (0, 7, 9)
OBS (0, 7, 10)
OBS (0, 7, 11)
OBS (1, 8, 1)
OBS (1, 8, 2)
OBS (1, 8, 3)
OBS (1, 8, 4)
OBS (1, 8, 5)
OBS (1, 8, 6)
OBS (1, 9, 1)
OBS (1, 9, 2)
OBS (1, 9, 3)
OBS (1, 9, 4)
OBS (1, 9, 5)
OBS (1, 7, 6)
OBS (1, 6, 6)
OBS (1, 5, 6)
OBS (1, 7, 7)
OBS (1, 7, 8)
OBS (1, 7, 9)
OBS (1, 7, 10)
OBS (1, 7, 11)


net1 (0, 5, 5) (0, 11, 3)
net2 (0, 5, 5) (0, 9, 10)
```

# Test case 4-high via penalty

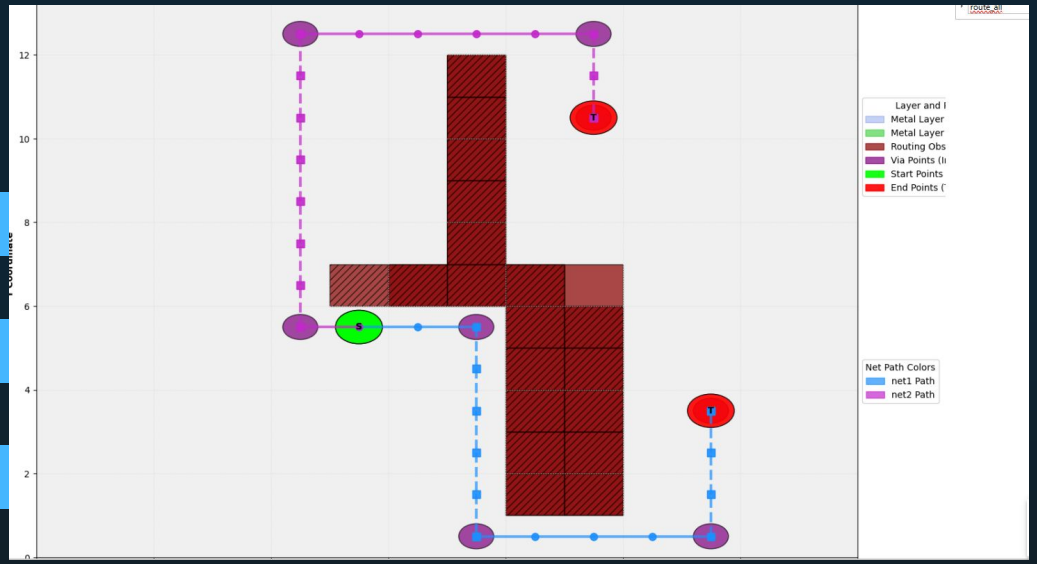



net1 routed with cost: 334.00
net2 routed with cost: 335.00

```
14, 14, 0, 80
OBS (0, 8, 1)
OBS (0, 8, 2)
OBS (0, 8, 3)
OBS (0, 8, 4)
OBS (0, 8, 5)
OBS (0, 8, 6)
OBS (0, 9, 1)
OBS (0, 9, 2)
OBS  (0, 9, 3)
OBS (0, 9, 4)
OBS (0, 9, 5)
OBS (0, 9, 6)
OBS (0, 7, 6)
OBS (0, 6, 6)
OBS  (0, 7, 7)
OBS (0, 7, 8)
OBS (0, 7, 9)
OBS (0, 7, 10)
OBS (0, 7, 11)
OBS (1, 8, 1)
OBS (1, 8, 2)
OBS (1, 8, 3)
OBS (1, 8, 4)
OBS (1, 8, 5)
OBS (1, 8, 6)
OBS (1, 9, 1)
OBS (1, 9, 2)
OBS (1, 9, 3)
OBS (1, 9, 4)
OBS (1, 9, 5)
OBS (1, 7, 6)
OBS (1, 6, 6)
OBS (1, 5, 6)
OBS (1, 7, 7)
OBS (1, 7, 8)
OBS (1, 7, 9)
OBS (1, 7, 10)
OBS (1, 7, 11)


net1 (0, 5, 5) (0, 11, 3)
net2 (0, 5, 5) (0, 9, 10)
```

# Test case 5- adding via and bend penalties

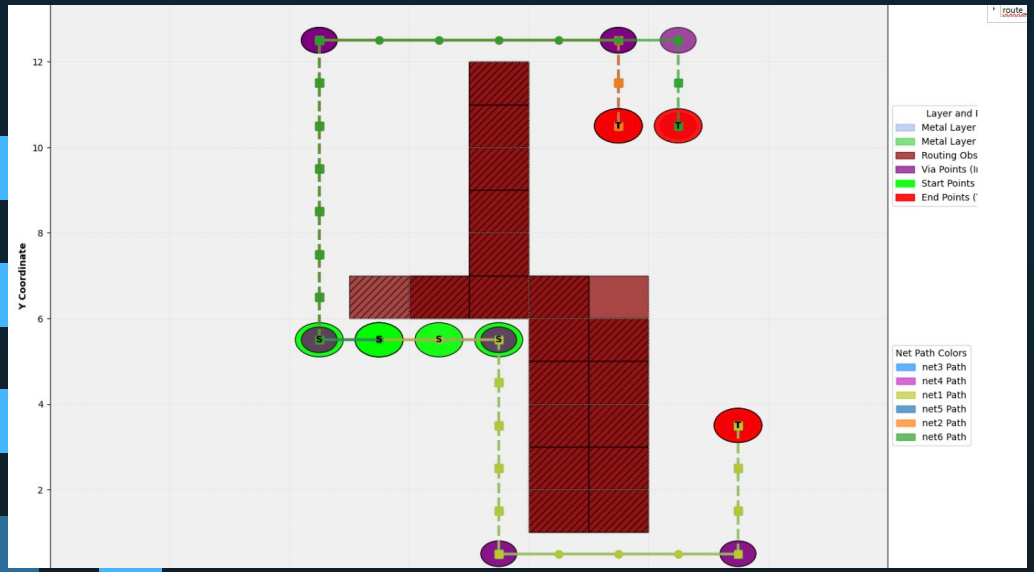



net1 routed with cost: 234.00
net2 routed with cost: 235.00

```
14, 14, 20, 20
OBS (0, 8, 1)
OBS (0, 8, 2)
OBS (0, 8, 3)
OBS (0, 8, 4)
OBS (0, 8, 5)
OBS (0, 8, 6)
OBS (0, 9, 1)
OBS (0, 9, 2)
OBS  (0, 9, 3)
OBS (0, 9, 4)
OBS (0, 9, 5)
OBS (0, 9, 6)
OBS (0, 7, 6)
OBS (0, 6, 6)
OBS  (0, 7, 7)
OBS (0, 7, 8)
OBS (0, 7, 9)
OBS (0, 7, 10)
OBS (0, 7, 11)
OBS (1, 8, 1)
OBS (1, 8, 2)
OBS (1, 8, 3)
OBS (1, 8, 4)
OBS (1, 8, 5)
OBS (1, 8, 6)
OBS (1, 9, 1)
OBS (1, 9, 2)
OBS (1, 9, 3)
OBS (1, 9, 4)
OBS (1, 9, 5)
OBS (1, 7, 6)
OBS (1, 6, 6)
OBS (1, 5, 6)
OBS (1, 7, 7)
OBS (1, 7, 8)
OBS (1, 7, 9)
OBS (1, 7, 10)
OBS (1, 7, 11)


net1 (0, 5, 5) (0, 11, 3)
net2 (0, 5, 5) (0, 9, 10)
```

# Test case 6- adding nets

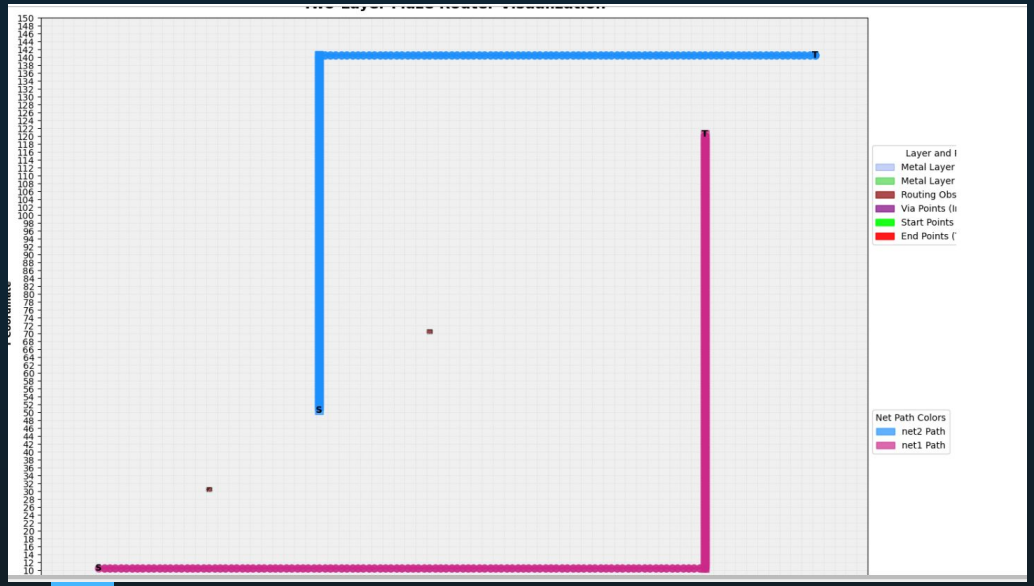



net1 routed with cost: 234.00
net2 routed with cost: 235.00

```
14, 14, 0, 0
OBS (0, 8, 1)
OBS (0, 8, 2)
OBS (0, 8, 3)
OBS (0, 8, 4)
OBS (0, 8, 5)
OBS (0, 8, 6)
OBS (0, 9, 1)
OBS (0, 9, 2)
OBS  (0, 9, 3)
OBS (0, 9, 4)
OBS (0, 9, 5)
OBS (0, 9, 6)
OBS (0, 7, 6)
OBS (0, 6, 6)
OBS  (0, 7, 7)
OBS (0, 7, 8)
OBS (0, 7, 9)
OBS (0, 7, 10)
OBS (0, 7, 11)
OBS (1, 8, 1)
OBS (1, 8, 2)
OBS (1, 8, 3)
OBS (1, 8, 4)
OBS (1, 8, 5)
OBS (1, 8, 6)
OBS (1, 9, 1)
OBS (1, 9, 2)
OBS (1, 9, 3)
OBS (1, 9, 4)
OBS (1, 9, 5)
OBS (1, 7, 6)
OBS (1, 6, 6)
OBS (1, 5, 6)
OBS (1, 7, 7)
OBS (1, 7, 8)
OBS (1, 7, 9)
OBS (1, 7, 10)
OBS (1, 7, 11)

net1 (0, 5, 5) (0, 11, 3)
net2 (0, 4, 5) (0, 9, 10)
net3 (0, 6, 5) (0, 11, 3)
net4 (0, 7, 5) (0, 9, 10)
net5 (0, 5, 5) (0, 9, 10)
net6 (0, 5, 5) (0, 10, 10)
```

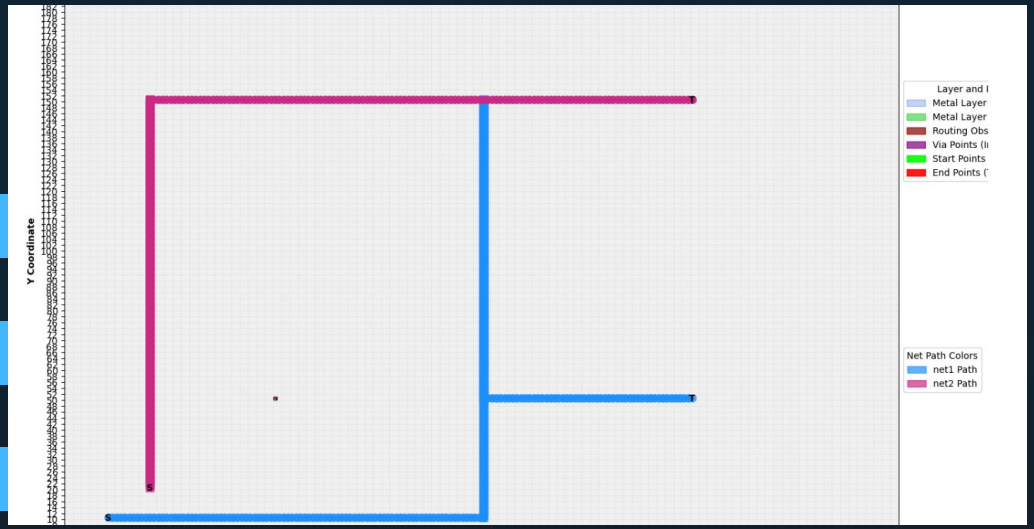
# Test case 7- uneven obstacle




```
150, 150, 20, 10
OBS (0, 70, 70)
OBS (1, 30, 30)
OBS (2, 100, 100)
net1 (0, 10, 10) (1, 120, 120)
net2 (1, 50, 50) (2, 140, 140)
```

net2 routing cost: 260.00
net1 routing cost: 270.00

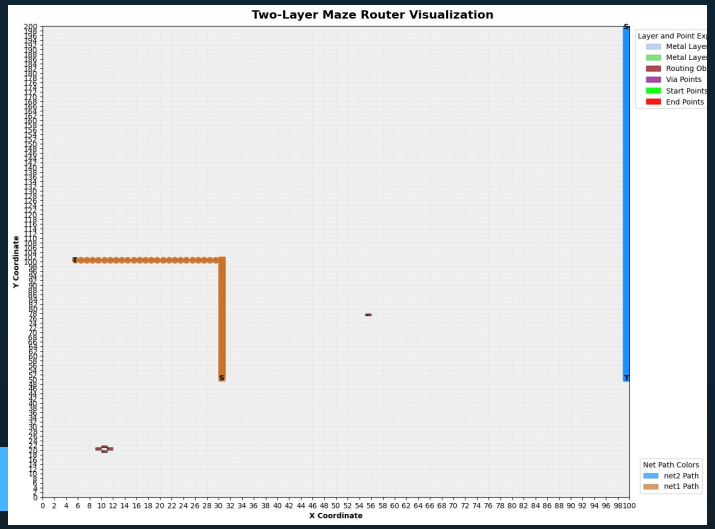
# Test case 8- multipin




```
200, 200, 30, 15
OBS (0, 100, 100)
OBS (1, 50, 50)
net1 (0, 10, 10) (1, 100, 150) (0, 150, 50)
net2 (1, 20, 20) (2, 150, 150) (1, 10, 150)
```

net1 routed with cost: 530.00
net2 routed with cost: 380.00

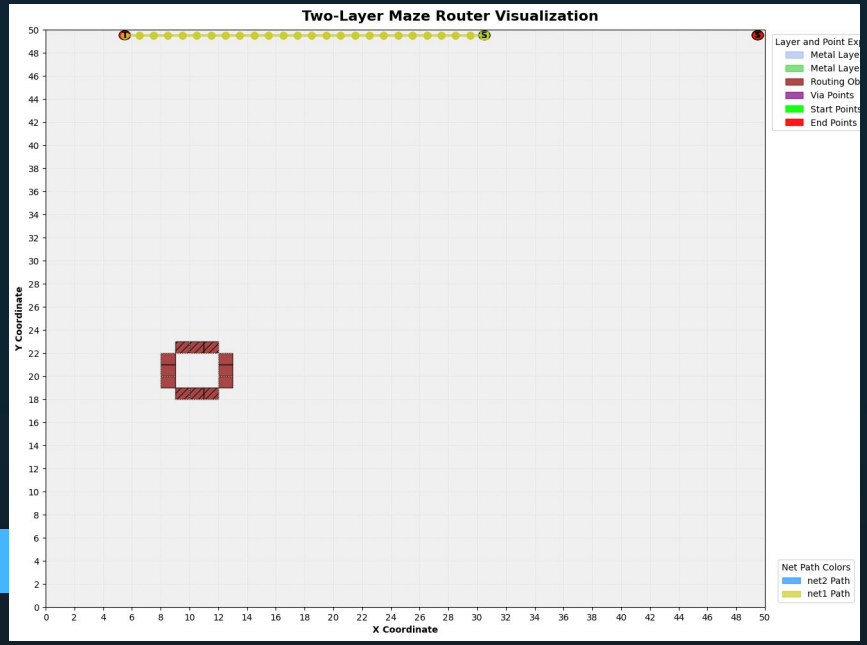
# Test case 9- multipin




```
100, 200, 20, 5
OBS (1, 10, 21)
OBS (1, 10, 19)
OBS (0, 9, 20)
OBS (0, 11, 20)
OBS (1, 55, 77)
net1 (1, 30, 50) (0, 10, 20) (0, 5, 100)
net2 (0, 99, 199) (0, 99, 50)
```

net2 routed with cost: 199.00
net1 routed with cost: 120.00

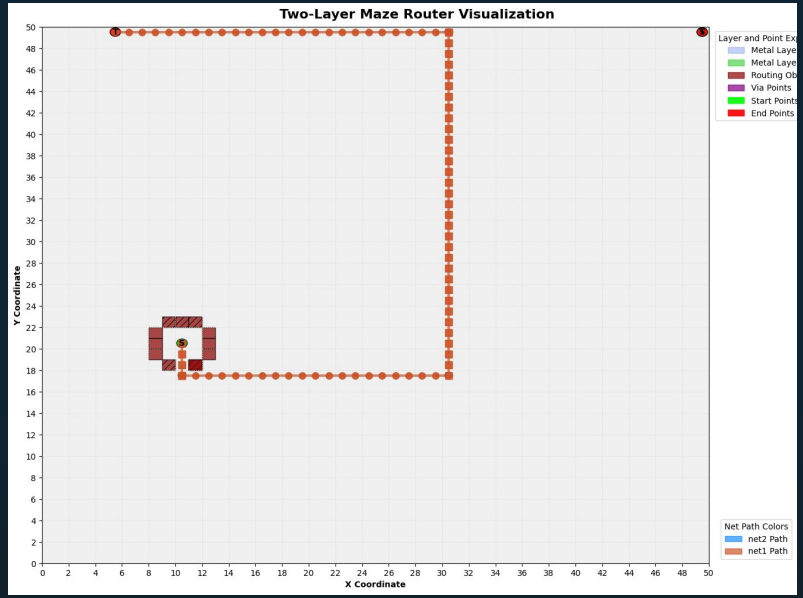
# Test case 10- Closed off Pin Afar




```
1   50, 50, 20, 5
2   OBS (1, 10, 22)
3   OBS (1, 9, 22)
4   OBS (1, 11, 22)
5   OBS (1, 10, 18)
6   OBS (1, 9, 18)
7   OBS (1, 11, 18)
8   OBS (0, 12, 20)
9   OBS (0, 12, 21)
10  OBS (0, 12, 19)
11  OBS (0, 8, 20)      You, 38 minutes ago •
12  OBS (0, 8, 21)
13  OBS (0, 8, 19)
14  OBS (1, 55, 77)
15  net1 (0, 10, 20) (1, 30, 50) (0, 5, 100)
16  net2 (0, 99, 199) (0, 99, 50)
17
```

net2 routed with cost: 0.00
net1 routed with cost: 50.00

# Test case 11- Does it escape?




```
50, 50, 20, 5
OBS (1, 10, 22)
OBS (1, 9, 22)
OBS (1, 11, 22)
OBS (1, 9, 18)
OBS (1, 11, 18)
OBS (0, 11, 18)
OBS (0, 12, 20)
OBS (0, 12, 21)
OBS (0, 12, 19)
OBS (0, 8, 20)
OBS (0, 8, 21)
OBS (0, 8, 19)
OBS (1, 55, 77)
net1 (0, 10, 20) (1, 30, 50) (0, 5, 100)
net2 (0, 99, 199) (0, 99, 50)
```

net2 routed with cost: 0.00
net1 routed with cost: 50.00

# Pins Number Case

- Nets
  ("net1", [(0, 2, 2), (0, 5, 5), (0, 11, 3), (0, 14, 6)])
  ("net2", [(0, 3, 12), (0, 8, 10), (0, 12, 8)])
  ("net3", [(0, 4, 4), (0, 10, 2), (0, 15, 7)])
  ("net4", [(0, 1, 13), (0, 7, 12), (0, 11, 9), (0, 15, 15)])

- Output

--- FIRST RUN: ROUTING WITHOUT NET PRIORITY ---
net1 routing cost: 136.00
net2 routing cost: 130.50
net3 routing cost: 184.00
net4 routing cost: 338.00

--- SECOND RUN: ROUTING WITH NET PRIORITY ---
net1 routing cost: 136.00
net4 routing cost: 202.50
net2 routing cost: 174.40
net3 routing cost: 230.00

# Manhattan Distance Case

- Nets
  ("net1", [(0, 0, 0), (0, 5, 5)])
  ("net2", [(0, 0, 0), (0, 8, 8)])
  ("net3", [(0, 0, 0), (0, 4, 4)])

- Output

--- FIRST RUN: ROUTING WITHOUT NET PRIORITY ---
net1 routing cost: 47.00
net2 routing cost: 80.25
net3 routing cost: 91.60

--- SECOND RUN: ROUTING WITH NET PRIORITY (Manhattan distance) ---
net3 routing cost: 45.00
net1 routing cost: 70.50
net2 routing cost: 106.80