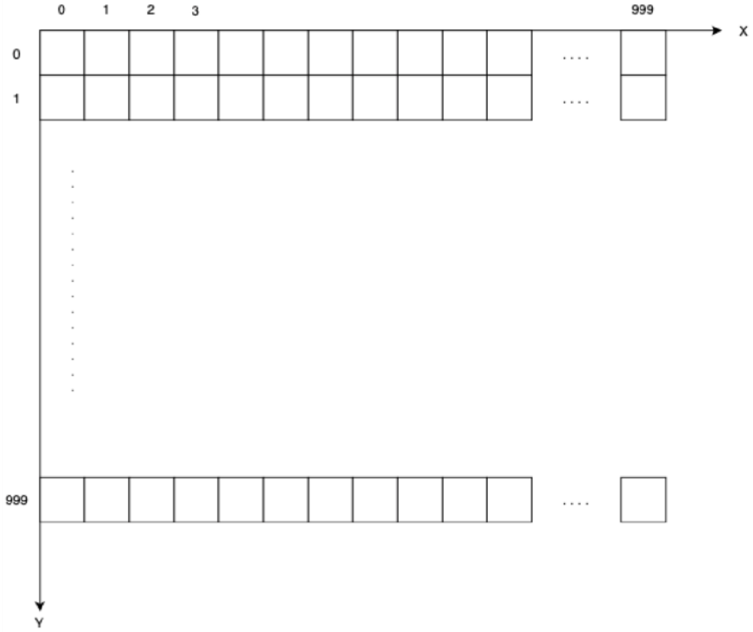
CSCE 3304 – Digital Design II

Project 2 (Bonus): Developing a Maze Router

This is a **bonus project worth 5%** of the digital design II coursework. In this project you will work in a **team** (same team as project 1 team) to develop a router that implements any maze routing algorithm (e.g., Lee's algorithm) using any programming language of your choice. The router connects pins that belong to the same net together using the available routing resources. There are two routing layers (M1: horizontal and M2: Vertical) and the routing grid is 1000x1000 cells for each layer.



The input to the router is a text file that lists the nets to be routed. Each line looks like the following

```
Net_name (pin_1_layer, pin_1_x, pin_1_y) (pin_2_layer, pin_2_x, pin_2_y) …
```

For example:

```
net1 (1, 10, 20) (2, 30, 50) (1, 5, 100)
net2 (2, 100, 200) (1, 300, 50)
net3 (1, 100, 50) (2, 300, 150) (2, 50, 50) (1, 2, 2)
.
.
```

The output from the router is a text file that lists the cells used by each net. Each line of this file looks like:

```
Net_name (cell_1_layer, cell_1_x, cell_1_y) (cell_2_layer, cell_2_x, cell_2_y) …
```

For example:

```
net1 (1, 10, 20) (1, 11, 20) (1, 12, 20) …
net2 (2, 100, 200) (2, 100, 201) (2, 100, 202) …
.
.
```

Your algorithm should minimize the usage of vias (to move between layers) by assigning a high cost to vias. You may route wires vertically on M1 and horizontally on M2 but do that only when there is no other choice (give a high cost for that).

Optionally as an **extra 1% bonus**, you can either (**pick one**):

- Support any number of routing layers instead of just 2. Please note, however, that the given test cases use 2 layers only. So, in case you implement this bonus, you will also need to create test cases to demonstrate this capability.
- Develop a script to visualize the output file to examine the routed nets.
- Add support for rip-up and re-route to your router when it fails to route some nets.

Deliverables

- **This project has a single milestone to be delivered through Blackboard by Saturday May 29th, 2021 11:59 PM**
- Your submission should be a single **compressed file** containing:
  - Documented source code
  - At least 8 test cases (Text files representing inputs and corresponding outputs).
  - A detailed **PDF** report documenting your project. The report should at least document your design, a user guide with screenshots, any extra assumptions you made, and any limitations and bugs you are aware of.
  - Note: the report must include a section indicating the role of each group member. Group members may receive different grades based on their contribution to the project.

Plagiarism is NOT tolerated and will be reported. **Late submissions will be not be accepted at all.**