National Taiwan University of Science and Technology
Department of Electrical Engineering

Algorithm Design and Analysis, Spring 2023

Programming Assignment #2

Detailed Routing for Photonic Integrated Circuits (PICs)

(due June 18, 2023 (Sunday) on-line)

1. Problem Description

This programming assignment asks you to write a detailed router for Photonic Integrated Circuits (PICs) that can route 2-pin nets (connection between two points) with waveguides on a grid plane. The major difference between metal wire routing for conventional ICs and wave guide routing for PICs is that waveguides can cross each other with additional waveguide loss. Given the size (the number of horizontal and vertical grids) of a routing grid plane, a netlist, the waveguide loss including propagation loss, crossing loss, and bending loss, the detailed router routes all nets in the routing region. The main objective is to minimize the total waveguide loss.

2. Input

The file format for the detailed routing problem is illustrated, with comments in italics (these will not be in actual input files). The 1st line gives the problem size in terms of the number of horizontal and vertical grids. The 2nd, 3rd, and 4th lines separately give the propagation loss, crossing loss, and bending loss. The 5th line gives the number of nets and following indicate each net, including starting position and terminal position. The input file format is as follows:

grid # # *// # horizontal tiles, # vertical tiles*
propagation loss α *// propagation loss of a net* $= \alpha \times wirelength\ \ of\ \ the\ \ net$
crossing loss β *// crossing loss of a grid* $= \beta \times \max(0, \#nets\ \ passing\ \ the\ \ grid - 1)$
bending loss γ *// bending loss of each wire bend*
num net # *// # nets*
net_id x_s y_s x_t y_t
...
// repeat for the appropriate number of nets

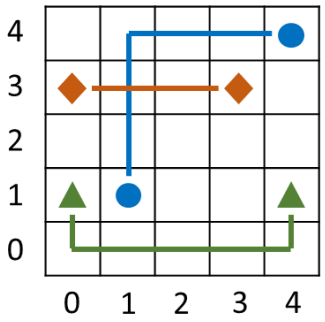
3. Output

All the routes in the output could only be horizontal lines and vertical lines. For example (18, 61)-(19, 62) is not acceptable, because it is diagonal. Remember that **each route could be different either in the x or y location only, and the difference must be 1.** The output file format is as follows:

[net_id] [# of routes, k]
[x_{11}] [y_{11}] [x_{12}] [y_{12}]
[x_{21}] [y_{21}] [x_{22}] [y_{22}]
...
[x_{k1}] [y_{k1}] [x_{k2}] [y_{k2}]
//repeat for the appropriate number of nets

Note that for a certain net, x_{11}, y_{11}, x_{k2} and y_{k2} must be the same as x_s, y_s, x_t and y_t in the input file respectively. Also, for any i, x_{i2} and y_{i2} must be the same as $x_{(i+1)1}$ and $y_{(i+1)1}$ respectively.

Sample case:



Sample input file:

```
grid 5 5
propagation loss 1
crossing loss 10
bending loss 3
num net 3
0 0 1 4 1
1 3 3 0 3
2 4 4 1 1
```

Sample output file:

```
0 6
0 1 0 0
0 0 1 0
1 0 2 0
2 0 3 0
3 0 4 0
4 0 4 1
2 6
4 4 3 4
3 4 2 4
2 4 1 4
1 4 1 3
1 3 1 2
1 2 1 1
1 3
3 3 2 3
2 3 1 3
1 3 0 3
```

The total waveguide length is 15, the total number of cross is 1, and the total number of bends is 3. Therefore, the total waveguide loss is $15 \times 1 + 1 \times 10 + 3 \times 3 = 34$.

4. Hints

You can first model the routing problem as a graph where each node represents a grid and each edge denotes the tile boundary between tiles. The cost of an edge could be set to reflect the usage of the grid by currently routed nets. Then this problem can be solved by Dijkstra's shortest path algorithm.

5. Language/Platform

(a) Language: C or C++.

(b) Platform: Unix/Linux.

6. Command-line Parameter

In order to test your program, you are asked to add the following command-line parameters to your program (e.g., picRouting.exe pic5x5.in pic5x5.out):

[executable file name] [input file name] [output file name]

7. Submission

You need to submit the following materials in a .tgz or a .zip file by following the naming rules highlighted in red (e.g., **B10807000-p2**.zip) at the course website by the deadline: (1) source codes (main.cpp), (2) a makefile to compile your programs, and (3) a report (**B10807000-p2**.docx) no more than 4 pages describing the idea, algorithms, and data structure your detailed router implements. Please check these items before your submission.

8. Grading Policy

This programming assignment will be graded based on (1) the correctness (a solution is correct if all nets are well-connected, i.e. no disconnection), (2) solution quality (determined by the total waveguide loss), (3) running time (the runtime is restricted in 10 minutes for each case, and the problem size is limited to 100x100 routing plane and 1000 nets), and (4) required submission files with correct file names. Please make sure that your program can be compiled and run in Linux before submission!

8. Online Resources

Sample input files (*.in), sample readme.txt, and sample makefile can be found at the course website.