

## Programming Assignment #1

# Open-Net Finder

Due: April 7

## 1 Problem Description

Engineering change orders (ECOs) are pervasively applied to modern physical design of nanometer integrated circuits for cost-effective design changes. After applying ECO, a net may become open, which results in a large number of disconnected net components. Each net component further consists of a set of connected net shapes and vias on different layers. It is very challenging to efficiently and effectively identify all disconnected net components and find an obstacle-avoiding minimal-cost routing path among those net components.

In this programming assignment, you are asked to implement the “open-net finder” for large-scale ECO routing. Given a net containing a set of  $n$  existing net shapes,  $N = \{N_1, N_2, \dots, N_n\}$ , and vias, and a set of  $l$  layers,  $L = \{L_1, L_2, \dots, L_l\}$ , the problem is to identify a set of  $c$  net components,  $C = \{C_1, C_2, \dots, C_c\}$ . The net is open if and only if the number of net components is more than one.

## 2 Input

Please refer to the contest website for the input format and all test cases, including the hidden ones: [https://iccad-contest.org/2017/Problem\\_B/default.html](https://iccad-contest.org/2017/Problem_B/default.html)

## 3 Output

The output file should include a set of  $c$  net components,  $C = \{C_1, C_2, \dots, C_c\}$ . The following table gives the output format and an output to the sample input. (Note that the solution may not be the optimal one.)

Output Format	Comments
<i>c</i>	// number of net components
C1	// net component #1
RoutedShape <Metal Layer> <x0, y0> <x1, y1>	// list all routed shapes and vias
RoutedVia <Via Layer> <x, y>	// of net component #1
C2	// net component #2
RoutedShape <Metal Layer> <x0, y0> <x1, y1>	// list all routed shapes and vias
RoutedVia <Via Layer> <x, y>	// of net component #2
⋮	⋮

## 4 Language/Platform

1. Language: C or C++ is preferred.
2. Platform: Linux.

## 5 Command-line Parameter

In order to test your program, you are asked to add the following command-line parameters to your program (e.g., FM input.dat output.dat):

./Lab1 [input file name] [output file name]

Please be noted that the name of your binary executable file must be “Lab1”.

## 6 Submission

You need to submit the following in a “tar” file to E3 (<https://e3.nycu.edu.tw/>) by the deadline. Please put all required files in a folder: (1) source codes, (2) Makefile, (3) a text readme file (readme.txt) stating how to build and use your program. The folder name must be your student ID. Be sure to compress the folder in the linux environment with the following command.

tar cvf Student\_ID.tar Student\_ID

## 7 Grading Policy

This programming assignment will be graded based on (1) the **correctness**, (2) **solution quality**, and (3) **running time**. For each case, the runtime limit is **1 hours**. It will be regarded as “failed” for the case if it takes more than 1 hours.

**There will be 20% penalty per day for late submission.**