Lab4 - Detailed routing project

Deadline: 23:55 Jan 2, 2022

Problem Statement

Implement a 2-layer detailed router to complete channel routing problems. You can use any rip-up and reroute based detailed router or greedy channel routing algorithm.

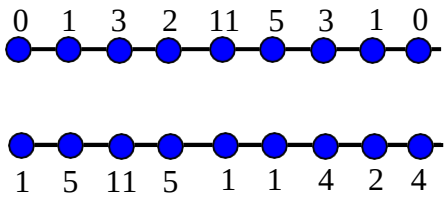
Input/Output Format

Input:

Input Example.

0 1 3 2 11 5 3 1 0

1 5 11 5 1 1 4 2 4



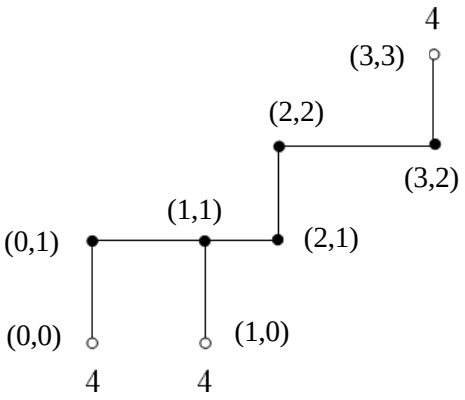
Three test cases, case1, case2, and Deutsch difficult are provided.

Output:

Text: print out all horizontal and vertical paths of every net in a file. Column's width and track's height is 1. Coordinate of bottom layer's first net is (0,0).

Format:

.begin net_name
.H lef_x lef_y rig_x
.V bot_x bot_ _y top_y
.end



Example. The above figure displays the routing of net 4 (three pins and 5 vias).

.begin 4

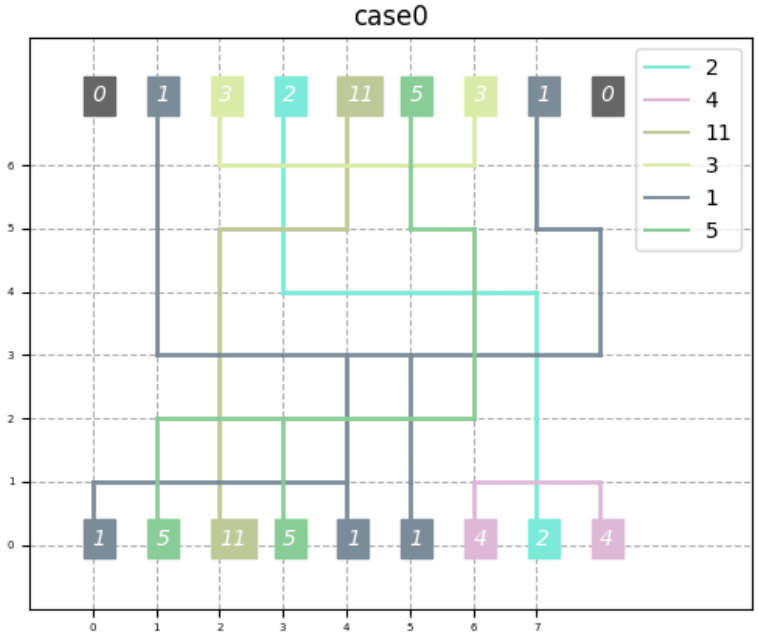
.H 0 1 2

.V 0 0 1

.V 1 0 1
.V 2 1 2
.H 2 2 3
.V 3 2 3
.end

Note that. There is no fixed wire segment order. A via is induced by the intersection of one horizontal and one vertical wire segment of the same net. If two wire segments of different nets with the same direction "overlap", a short error occurs. You can use the verifier to examine the routing errors.

Graphic (optionally): You can draw the routing results on the screen to make debugging easier.



Evaluation

1. You **MUST WRITE YOUR OWN CODE**. Copying codes may result in you **failing** thiscourse.
2. For each case, the run time limit is set up to **3 minutes**. It will be regarded as "failed"if you use more than 3 minutes.
3. Naming rule.
   A. Name of the binary after "make" – Lab4
   B. If not following specified naming rule, you will receive zero mark
3. Late submissions will **NOT** be accepted.

Executing Procedure

1. **./Lab4 [input.txt] [output.txt]**
2. Search for [output.txt], if not found→ break → 0 point
3. **./ verifier [output.txt] [input.txt]**
4. If fail → break→ 0 point

Ranking

A. Can run small case but fail in Deutsch difficult (cannot generate

routing result) – 60

B. Ranking is mainly based on the correctness of routing results and the required number of tracks to complete the routing of "Deutsch difficult.txt".

If the routing result of "Deutsch difficult.txt" is correct, ranking is categorized into following sets:

Maximum tracks	Minimum tracks	Base score
infinity	51	75
50	41	80
40	31	85
30	25	90
24	22	95
1	19	100

C. No spill-over area is allowed. (No extra columns off the edge.)

D. For each case, the run time limit is up to 180 seconds. It will be regarded as "failed" if you use more than 180 seconds.

Environment

1. Linux (Please make sure your code is available on our linux server(compile &execute). If it does not work, you will get zero points!!)
2. Makefile should be provided

Submission

<1>

Please put all required files in a folder, the name of the folder is your Student_ID.

```
Student_ID
    |__Source code (.cpp, .h)
    |__Makefile
```

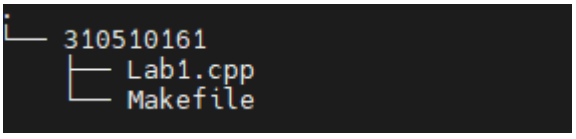
<2>

Use below command to compress the folder in the linux environment, and thecompressed file name should as same as your Student_ID.

```
tar cvf Student_ID.tar Student_ID
```

Example

<1>



<2>

```
21:19 zu00895077@vda04 [~/PDA_TA/Lab1] >$ ls
310510161/
21:19 zu00895077@vda04 [~/PDA_TA/Lab1] >$ tar cvf 310510161.tar 310510161
310510161/
310510161/Lab1.cpp
310510161/Makefile
21:21 zu00895077@vda04 [~/PDA_TA/Lab1] >$ ls
310510161/  310510161.tar
```

Upload Student_ID.tar to e3
(If your submission file is not .tar, you will get zero point!