

Lab1 – Channel routing

Deadline: 3/27/2023 23:59

Problem description

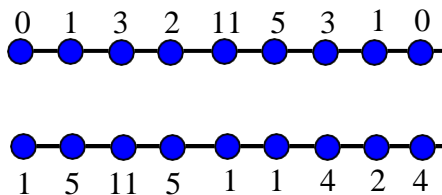
Implement a 2-layer detailed router to complete channel routing problems. You need to use the greedy channel routing algorithm.

Input/Output Format

Input:

The input file consists of two rows of integers, where each number represents the net id of the pin except for 0. All the pins located at the top and bottom rows with the same net id must be connected together.

Example:

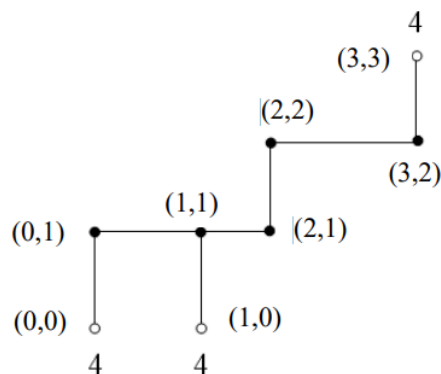


Output:

Print out all horizontal and vertical segments of every net in a file. The column's width and track's height are 1. The coordinate of the bottom layer starts from 0.

Format:

```
.begin net_name  
.H left_x left_y right_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, and we set **via cost = 5**. 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. Besides, please notice that the top and bottom rows can't place any horizontal wire segments (for example, ".H 0 0 1" is illegal in the above case).

Appendix

Verifier.

We provide a verifier to check the correctness of your program.

```
./verifier [output.txt] [input.txt]
```

If the terminal shows "All signals are connected successfully. ", it means your program passes the verifier. Then it will also show your track count and total wire length(including via cost), which are the basis of our grading.

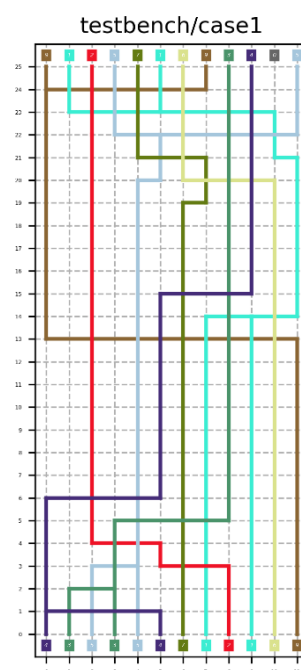
Plotter.

We provide a Python plotter that can help you debug from the visualized routing result.

```
python plotter.py --in_file input_filename --out_file output_filename --img_name img_name
```

You can execute the following command to see an example result of case1

```
python plotter.py --in_file testbench/case1.txt --out_file result.txt --img_name example
```



Notice that. The final grade is based on the output message of the **verifier**, the Python plot is only for reference.

Makefile.

We also provide a Makefile to help you build your code. You can also write your own Makefile as long as it can be executed by the "make" command and generate a

specified executable.

Executing Procedure

1. We'll use "make" command to build your code, please make sure that your Makefile can generate an executable named **Lab1**
2. **./Lab1 [input.txt] [output.txt]**
3. Search for [output.txt], if not found → break → 0 point
4. **./verifier [output.txt] [input.txt]**
5. If fail → break → 0 point

Submission

Please submit the following materials in a .zip file to E3 by the deadline, specifying your student ID in the subject field (e.g., StudentID.zip):

- (1) Source codes (.cpp, .h ...)
- (2) Makefile

Grade

Notice that we provide several testcases, but your score is evaluated only based on the output result of testcase "**Deutsch_difficult.txt**", and the grading rule is listed below:

- A. For each case, the runtime limit is up to **300 seconds**. It will be regarded as "failed" if you use more than 300 seconds.
- B. Spill-over area is allowed, but you will get an additional penalty according to the width of spill-over region.
penalty = -(the width of spill-over region)
- C. Ranking is categorized into the following sets according to your track number of "Deutsch_difficult.txt". **You'll get 100 if your track number is 19**. Notice that if two or more students have the same track number (except 19), we'll compare your wire length to determine who will get a higher score.

Maximum tracks	Minimum tracks	Score region
infinity	53	[70, 75)
53	44	[75-80)
43	34	[80-85)
33	28	[85-90)
27	25	[90-95)
24	20	[95-100)
19		100

Suppose your program can generate a correct routing result of "Deutsch_difficult.txt" in 300 seconds, and there are **n** students in the same score region **r**, then your score should be

$$\text{Score} = \min(r) + \max\left(0, 5 \times \frac{n+1-\text{your_WL_rank_within_n_students}}{n}\right) + \text{penalty}$$

- D. **Late submission:** Score * 8 before 4/3. After 4/3, you will get 0 point
- E. **Wrong submission format:** 10 points punishment
- F. **Any work by fraud will absolutely get 0 point**