

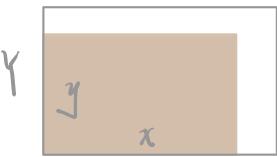
Programming Assignment #2: Fixed-outline Floorplanning (due 6pm, May 3, 2020 on-line)

Modified from Problem #1 of the 2003 IC/CAD Contest (Source: Springsoft/Synopsys);
 please see also Exercise 10.8 of the W&C&C book (pages 628-630).

$$0.2 \times 0.4 () + 0.6 () = 0.8$$

1. Problem Statement

This programming assignment asks you to write a fixed-outline chip floorplanner that can handle hard macros. Given a set of rectangular macros and a set of nets, the floorplanner places all macros within a rectangular chip without any overlaps. We assume that the lower-left corner of this chip is the origin $(0,0)$, and no space (channel) is needed between two different macros. The objective is to minimize the area of chip bounding box and the total net wirelength. The total wirelength W of a set N can be computed by



$$W = \sum_{n_i \in N} HPWL(n_i), \quad \beta \left(\alpha \frac{A_{\text{outline}}}{\text{outline}} + (1-\alpha) \frac{W}{W_{\text{outline}}} \right) + (1-\beta) \left(0.5 \frac{A_{\text{outline},w}}{\text{outline},w} + 0.5 \frac{A_{\text{outline},h}}{\text{outline},h} \right)$$

where n_i denotes a net in N , and $HPWL(n_i)$ denotes the half-perimeter wirelength of n_i .

The objective for this problem is to minimize

$$\alpha (0.5x + 0.5y) + (1-\alpha) \frac{W}{W_{\text{norm}}}$$

$$\checkmark \quad Cost = \underbrace{\alpha \frac{A}{A_{\text{norm}}} + (1-\alpha) \frac{W}{W_{\text{norm}}}}_{0.7} + (1-\beta) (R^* - R) \quad \beta$$

where A denotes the bounding-box area of the floorplan, A_{norm} is the average area, W is the total wire length, W_{norm} is the average wire length, and α , $0 \leq \alpha \leq 1$, is a user defined ratio to balance the final area and wirelength. To compute A_{norm} and W_{norm} , we can perturb the initial solution or m times to obtain m floorplans and compute the average area A_{norm} and the average wire length W_{norm} of these floorplans. The value m is proportional to the problem size. Note that a floorplan which cannot fit into the given outline is not acceptable.

$$\Phi(F) = \alpha A + \beta W + (1-\alpha-\beta)(R^* - R)^2$$

	Chip area	Wirelength	Aspect ratio penalty
A	Chip area		
α	Area weight		
W		Wirelength	
β		Wirelength weight	
R^*			Desired aspect ratio
R			Current floorplan aspect ratio

2. Input/Output Specification

Input Format

$$\alpha A + \beta W$$



$$\text{perturb}(100) \rightarrow \frac{A_{\text{norm}}}{W_{\text{norm}}}$$



Each test case has two input files, *input.block* and *input.nets*. The first file (*input.block*) gives the outline size, the number of blocks, and the number of terminals defined in this file. Then the block dimensions are listed, followed by the terminal locations. The file format is as follows:

```

Outline: <outline width, outline height>
NumBlocks: <# of blocks>  $\sqrt{23} = 5$ 
NumTerminals: <# of terminals>

<macro name> <macro width> <macro height>
... More macros

<terminal name> terminal <terminal x coordinate> <terminal y coordinate>
... More terminals
  
```

The second file gives the number of nets in the floorplan, followed by the terminal information for each net. The file format is as follows:

```

NumNets: <# of nets>
NetDegree: <# of terminals in this net>
<terminal name>
... More terminal names

... More “NetDegree” and “terminal name”
  
```

The user-defined ratio α is given through the command-line argument. It ranges between 0 and 1.

Output Format

The output file (*output.rpt*) records the problem output. This report consists of six parts: (1) the final cost, (2) the total wirelength, (3) the chip area, (4) the chip width and height, (5) the runtime in seconds, and (6) the bounding-box coordinate for

```

<final cost>
// Cost =  $\alpha A + (1-\alpha)W$ 
<total wirelength>
//  $W = \sum_{n_i \in N} HPWL(n_i)$ 
<chip_area>
// area = (chip_width) * (chip_height)
<chip_width><chip_height>
// resulting chip width and height
<program_runtime>
// report the runtime in seconds
<macro_name>    <x1>    <y1>    <x2>    <y2>
<macro_name>    <x1>    <y1>    <x2>    <y2>
// (x1, y1): lower-left corner, (x2, y2): upper-right corner
... More macros
  
```

each macro (specified by the lower-left corner and upper-right corner). The report file format is shown above.

3. Command-line Parameters

The executable binary file must be named as “**fp**” and use the following command format:

```
./fp [ $\alpha$  value] [input.block name] [input.net name] [output file name]
```

For example, if you would like to run your binary for the input file **input.block** and **input.nets** with $\alpha = 0.5$ and generate a solution named **output.rpt**, the command is as follows:

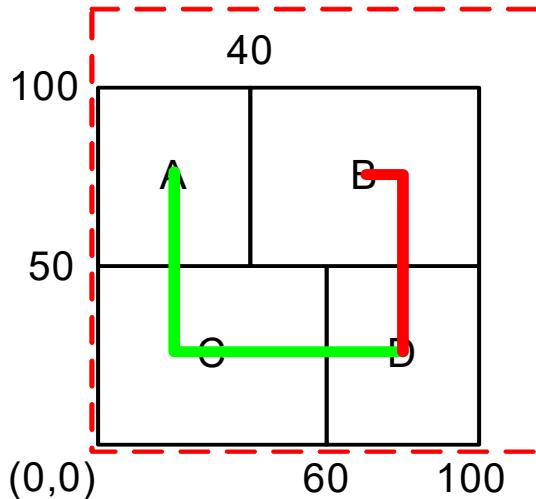
```
./fp 0.5 input.block input.nets output.rpt
```

4. Example

Figure 1 illustrates an example of the IO files (assume $\alpha = 0.5$):

Input files (input.block):

Outline: 120 120
NumBlocks: 4
NumTerminals: 0
A 40 50
B 60 50
C 60 50
D 40 50



(input .nets)

NumNets: 2
NetDegree: 3
A
C
D
NetDegree: 2
B
D

5085
170
10000
100 100
0.24
A 0 50 40 100
B 40 50 100 100
C 0 0 60 50
D 60 0 100 50

Output files (output.rpt)

5. Language/Platform

- Language: C/C++
- Platform: Linux. Please also develop your programs on the servers in the EDA Union Lab.

6. Advanced Features

Provide a GUI (Graphic User Interface) to show the floorplanning result.

7. Submission:

You need to create a directory named <student id>_pa2/ (e.g. f08943000_pa2/) which must contain the following materials:

- A directory named src/ containing your source codes: only *.h, *.hpp, *.c, *.cpp are allowed in src/, and no directories are allowed in arc/;
- A directory named bin/ containing your executable binary named fp;
- A makefile name makefile or Makefile that produces an executable binary from your source codes by simply typing “make”: the binary should be generated under the directory <student id>_pa2/;
- A text readme file named readme.txt describing how to compile and run your program.
- A report named report.pdf on the data structures used in your program and your findings in this programming assignment.

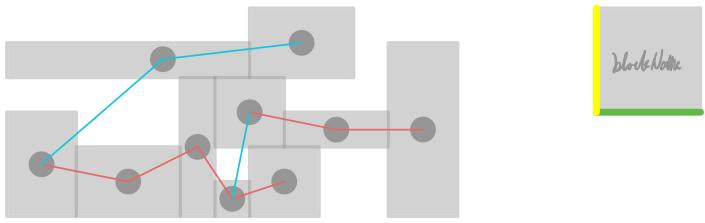
8. Evaluation

This programming assignment will be graded based on the (1) correctness of the program, (2) solution quality, (3) running time, (4) report.doc, and (5) readme.txt. Please check these items before your submission. Note that a floorplanning result will not be graded if the running time exceeds one hour on a machine in the EDA Union Lab.

9. Online Resources

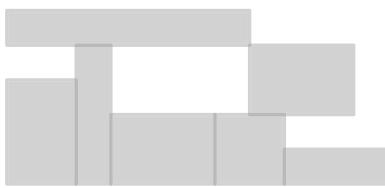
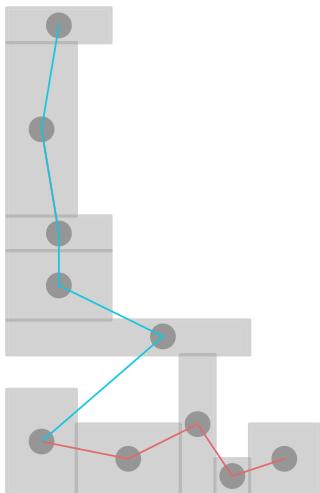
Sample input files (ami33.block/ami33.nets), readme.txt, and report.doc can be found at the NTU COOL course link below:

<https://cool.ntu.edu.tw/courses/1331/assignments/10272>



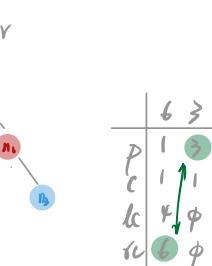
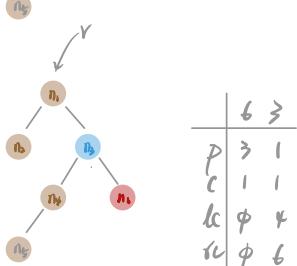
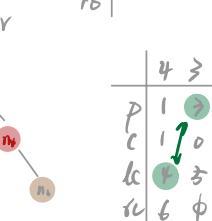
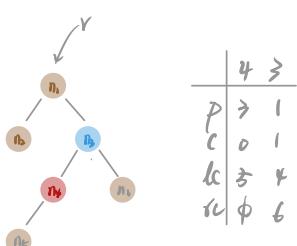
$b^*_tree_block \ \chi(\text{input_blocks})$:

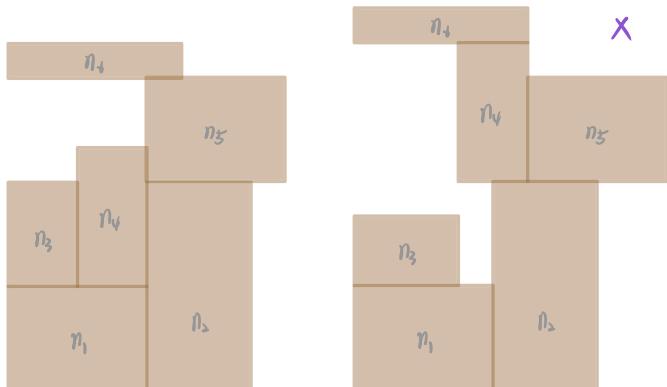
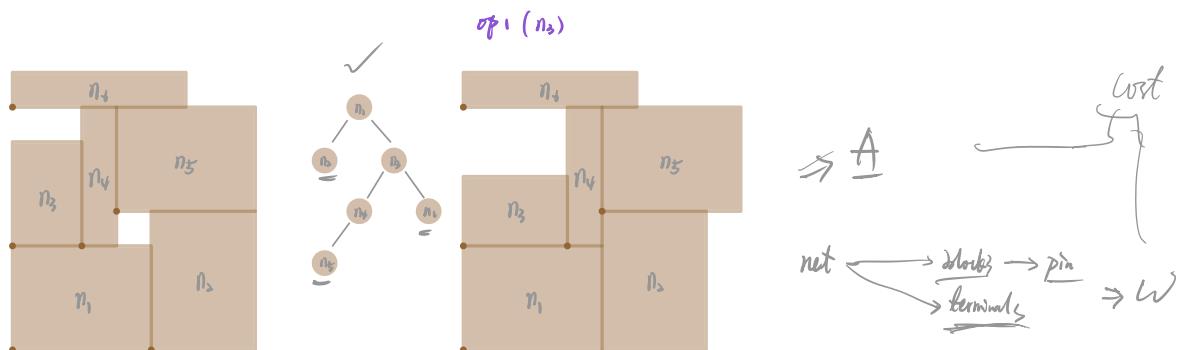
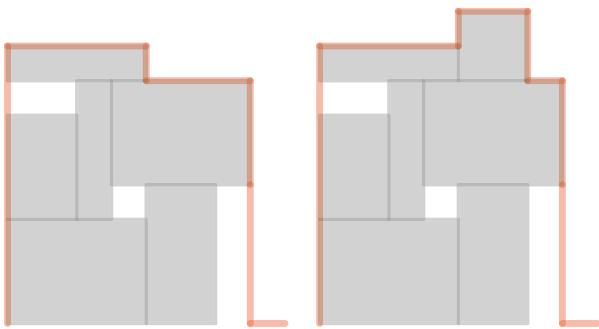
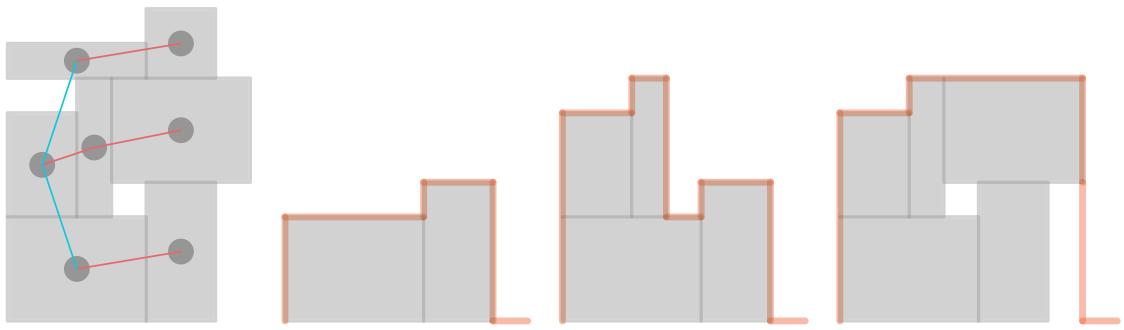
net \longleftrightarrow block \rightarrow pin
 \longrightarrow terminal



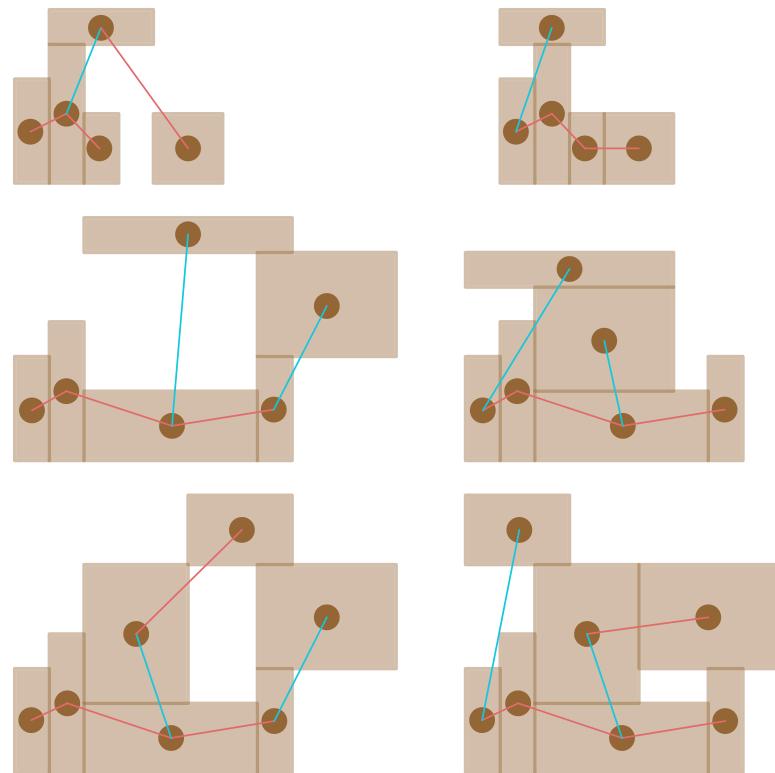
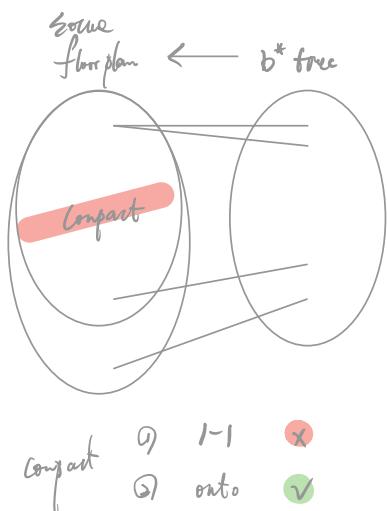
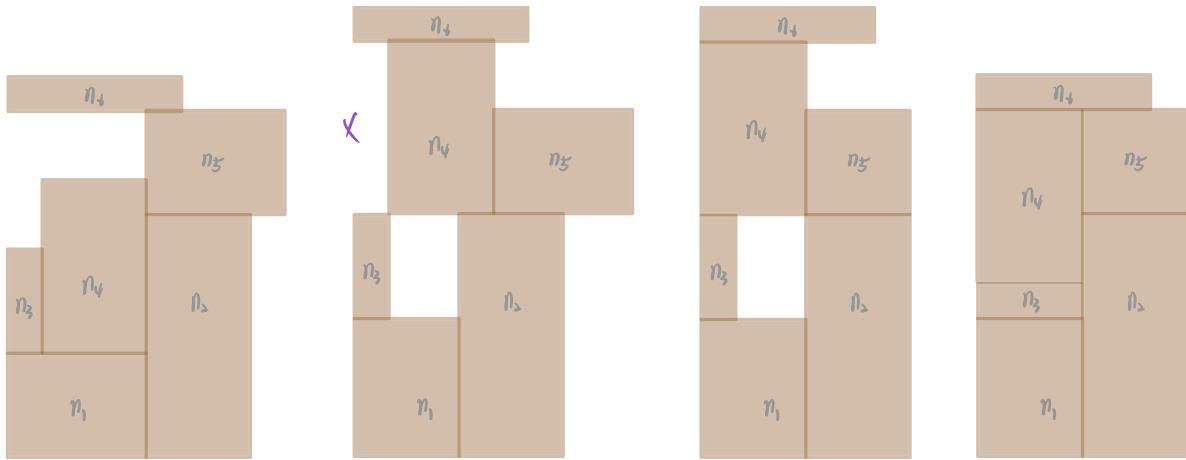
blocks	name	area	coor
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

free	to	to	χ
0	1	5	
1	2		D
2	3		1
3	4		2
4			3
5			
6	0		
7	7		
8	8		
9	9		
10			



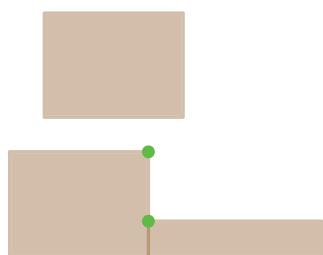


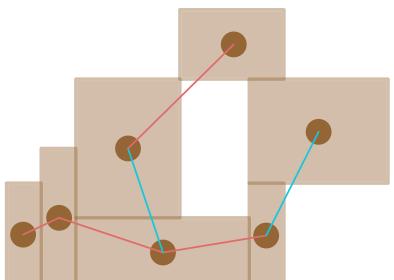
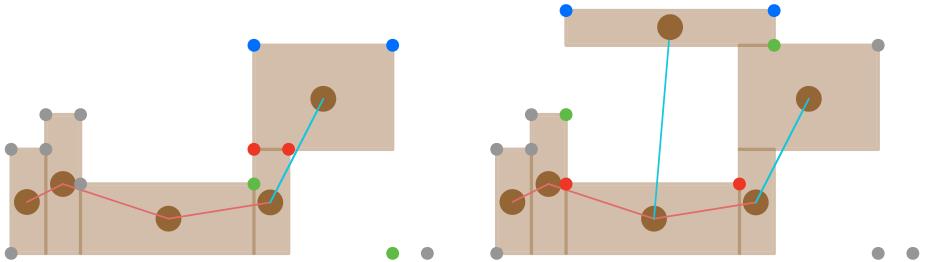
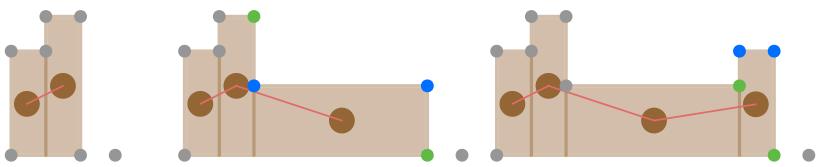
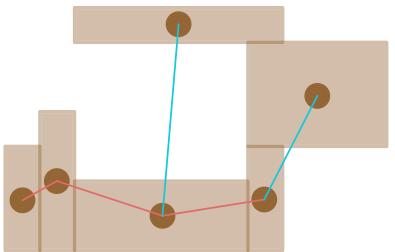
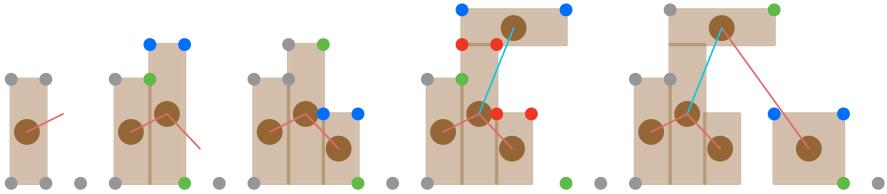
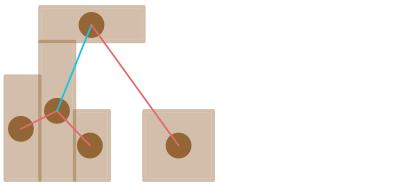
ID-block (name)
(terminal (name))
blocks
terminals
D-star-tree
nets

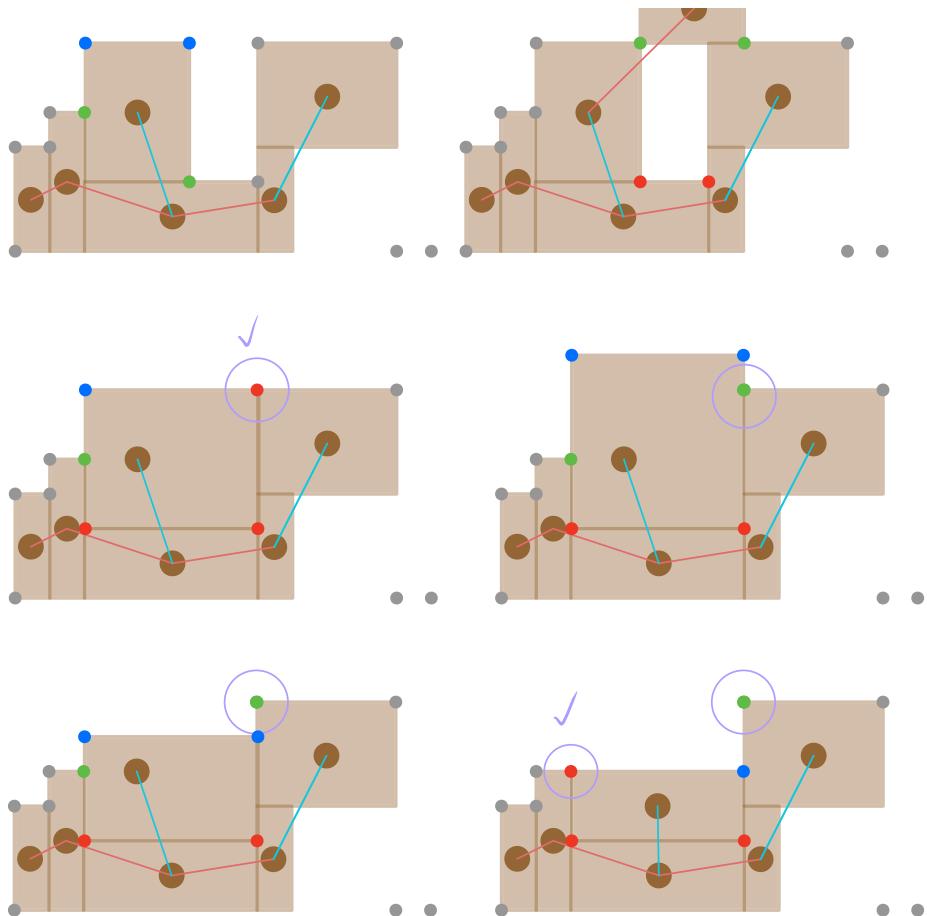


not compact $\xrightarrow{\text{向左, 向下}} \text{compact}$

b
1
>
3
4
5







1. 我去[A]否同(念)

行將有影

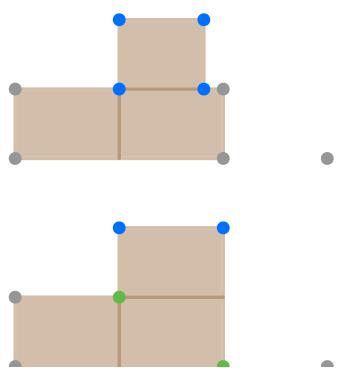
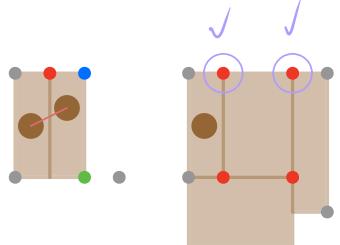
2. 留前後件點，中間刪掉，並加入產點

3. 第一顆重複則刪除二顆



4. 最後一顆向右拉

最後一顆重複則刪除二顆



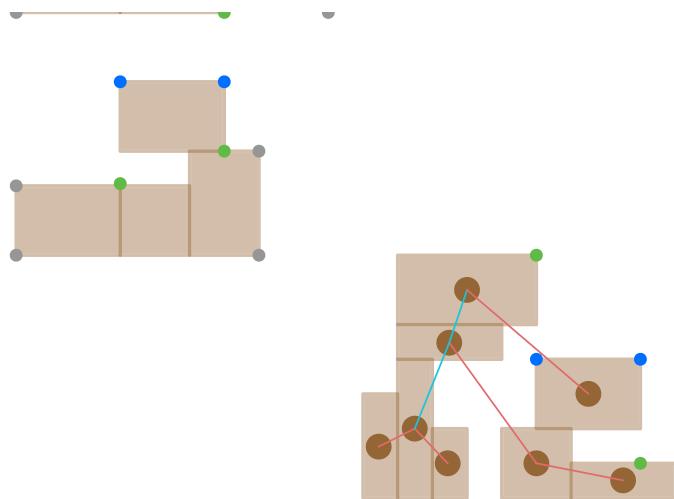
步
加四產

步有

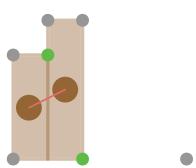
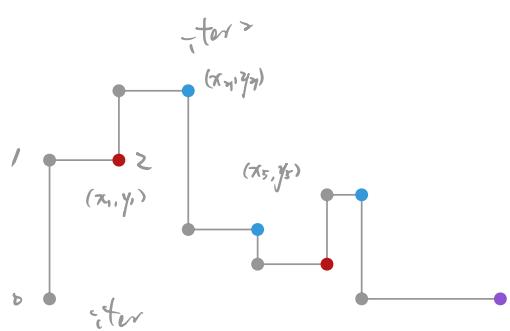
左右拉

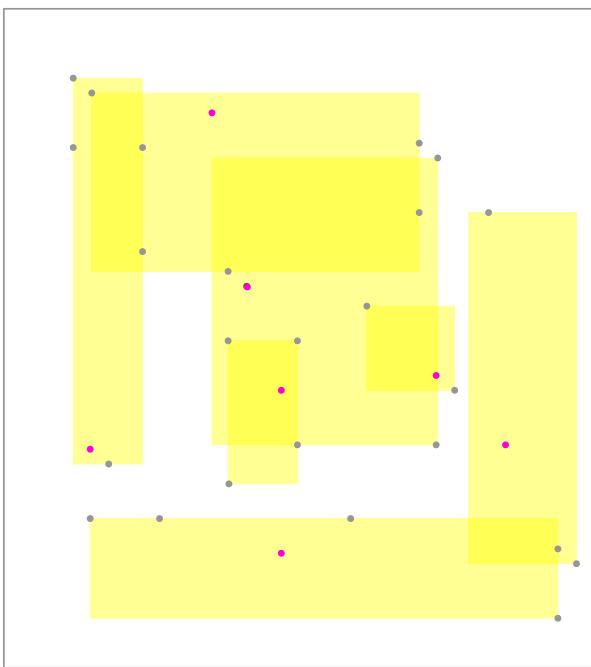
刪江加產

一樣則刪二



$y_7(x_7 - x_1)$

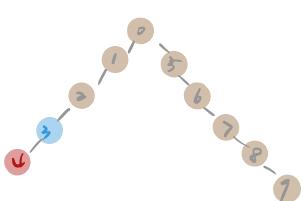
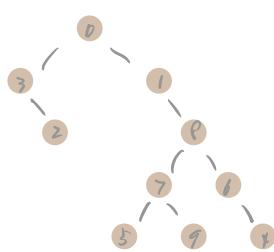




$\text{net} \rightarrow$ terminal Coor
 Block Center

~~down?~~ ~~map?~~
colored-list

tree	z	y	x
0	3	1	
1	8	0	
2	3		
3	2	0	
4	6		
5	7		
6	4	8	
7	5	9	8
8	7	6	1
9			7
10			



tree	z	y	x
0	1	5	

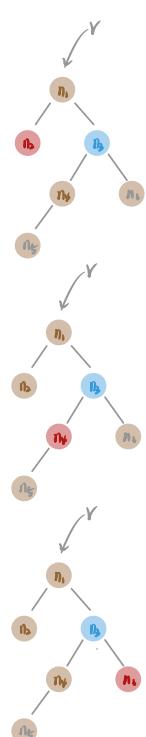
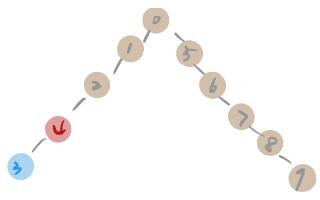
tree	z	y	x
0	1	5	

tree	z	y	x
0	1	5	

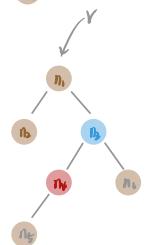
1	2	0
2	3	1
3	4	2
4	5	3
5	6	0
6	7	2
7	8	6
8	9	7
9		8
10		

1	2	0
2	3	1
3	4	2
4	5	3
5	6	0
6	7	2
7	8	6
8	9	7
9		8
10		

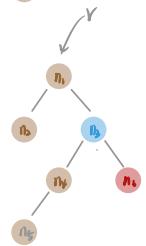
1	2	0
2	3	1
3	4	2
4	5	3
5	6	0
6	7	2
7	8	6
8	9	7
9		8
10		



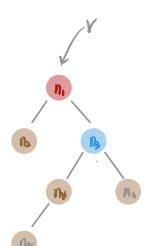
lc	4	x
rc	6	x
p	1	1
c	1	v



lc	4	5
rc	6	x
p	1	3
c	1	0



lc	4	x
rc	6	x
p	1	3
c	1	1



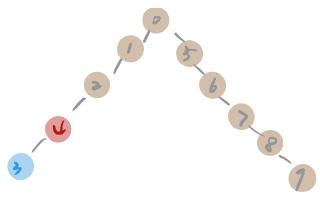
lc	4	2
rc	6	3
p	1	x
c	1	x



lc	2	4
rc	3	6
p	x	1
c	x	1

rotate
swap
move_L
move_R

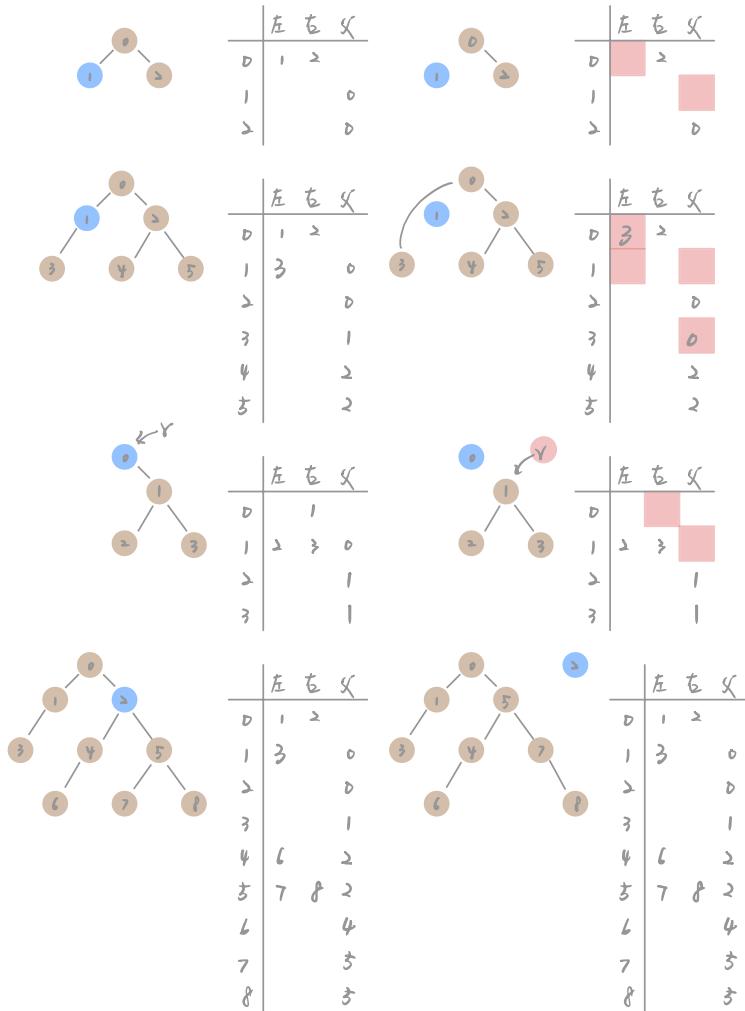
$$ID-1 \neq ID-2$$



- Case 1: A leaf node. We can simply delete the target leaf node.
- Case 2: A node with one child. We remove the target node and then place its only child at the position of the removed node. The tree update can be performed in $O(1)$ time.
- Case 3: A node with two children. We replace the target node n_t by either its right child or its left child n_c . Then we move a child of n_c , if any, to the original position of n_c . This process proceeds until the corresponding leaf node is handled. It is obvious that such a deletion operation requires $O(b)$ time, where b is the height of the B*-tree.

We can insert a new node into either an internal or an external position as follows.

- Internal position:** A position between two nodes in a B*-tree.
- External position:** A position pointed by a NULL pointer. Each node has two pointers, the left child and the right child. When the node has no left or right child, it points NULL.

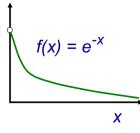


$$\text{Prob}(S \rightarrow S') = \begin{cases} 1 & \text{if } \Delta C \leq 0 \\ e^{-\frac{\Delta C}{T}} & \text{if } \Delta C > 0 \end{cases} \quad / * \text{"down-hill" moves} */ \quad / * \text{"up-hill" moves} */$$

- $\Delta C = \text{cost}(S') - \text{Cost}(S)$
- T : Control parameter (temperature)
- Annealing schedule: $T = T_0, T_1, T_2, \dots$, where $T_j = r^j T_0$, $r < 1$.

Unit 4

Y.-W. Chang



9

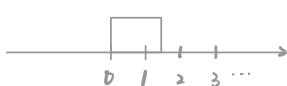
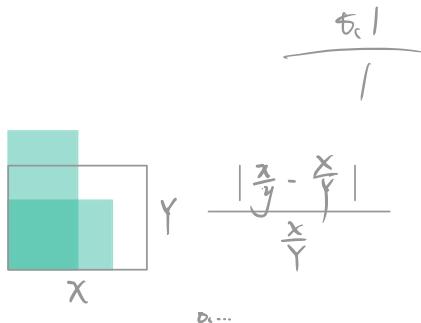
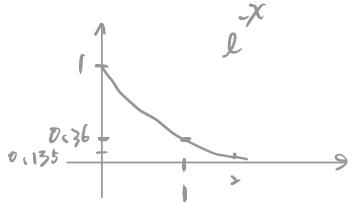
Generic Simulated Annealing Algorithm

```

1 begin
2 Get an initial solution  $S$ ;
3 Get an initial temperature  $T > 0$ ;
4 while not yet "frozen" do
5   for  $1 \leq i \leq P$  do
6     Pick a random neighbor  $S'$  of  $S$ ;
7      $\Delta \leftarrow \text{cost}(S') - \text{cost}(S)$ ;
    /* downhill move */
8     if  $\Delta \leq 0$  then  $S \leftarrow S'$ 
    /* uphill move */
9     if  $\Delta > 0$  then  $S \leftarrow S'$  with probability  $e^{-\frac{\Delta}{T}}$ 
10     $T \leftarrow rT$ ; /* reduce temperature */
11  return  $S$ ;
12 end

```

get [100%]



1. Problem Statement

This programming assignment asks you to write a fixed-outline chip floorplanner that can handle hard macros. Given a set of rectangular macros and a set of nets, the floorplanner places all macros within a rectangular chip without any overlaps. We assume that the lower-left corner of this chip is the origin (0,0), and no space (channel) is needed between two different macros. The objective is to minimize the area of chip bounding box and the total net wirelength. The total wirelength W of a set N can be computed by

$$W = \sum_{n_i \in N} HPWL(n_i) = \sum_{n_i \in N} \left(\alpha \frac{A_{\text{outline}} + (-\alpha) \frac{W}{W_{\text{outline}}}}{W_{\text{outline}}} + (1-\alpha) \left(\frac{A_{\text{outline}}}{W_{\text{outline}}} + 0.5 \frac{A_{\text{outline}}}{W_{\text{outline}, b}} \right) \right)$$

where n_i denotes a net in N , and $HPWL(n_i)$ denotes the half-perimeter wirelength of n_i .

The objective for this problem is to minimize

$$\text{Cost} = \alpha \frac{A}{A_{\text{norm}}} + (1-\alpha) \frac{W}{W_{\text{norm}}} + (1-\beta) (R^* - R)$$

where A denotes the bounding-box area of the floorplan, A_{norm} is the average area, W is the total wire length, W_{norm} is the average wire length, and $\alpha, 0 \leq \alpha \leq 1$, is a user defined ratio to balance the final area and wirelength. To compute A_{norm} and W_{norm} , we can perturb the initial solution m times to obtain m floorplans and compute the average area A_{norm} and the average wire length W_{norm} of these floorplans. The value m is proportional to the problem size. Note that a floorplan which cannot fit into the given outline is not acceptable.

$$\Phi(F) = \alpha A + \beta W + (1-\alpha-\beta)(R^*-R)^2$$

2. Input/Output Specification

Input Format

$$\alpha A + \beta W$$

A	Chip area
α	Area weight
W	Wirelength
β	Wirelength weight
R^*	Desired aspect ratio
R	Current floorplan aspect ratio

$$\text{Cost} = 1$$

$$T \leq 1$$

$$\frac{0.95}{0.95}$$

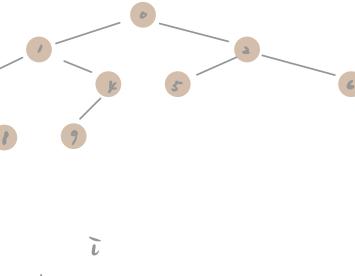
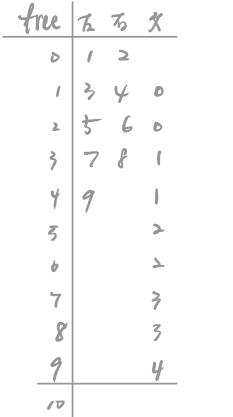
$$T \leq 1$$

$$0.3$$



$$b \sim 1$$

$$1-$$

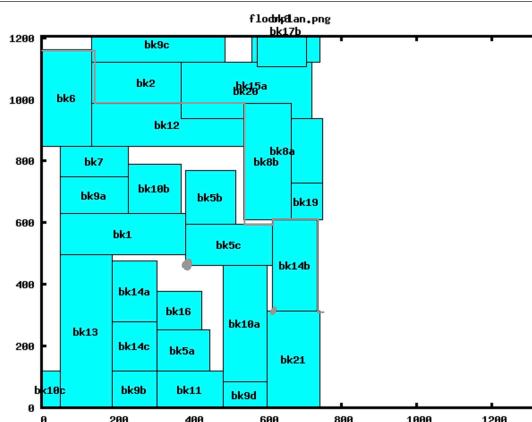


$$2^{i+1} \geq i+2$$

602f

143

11

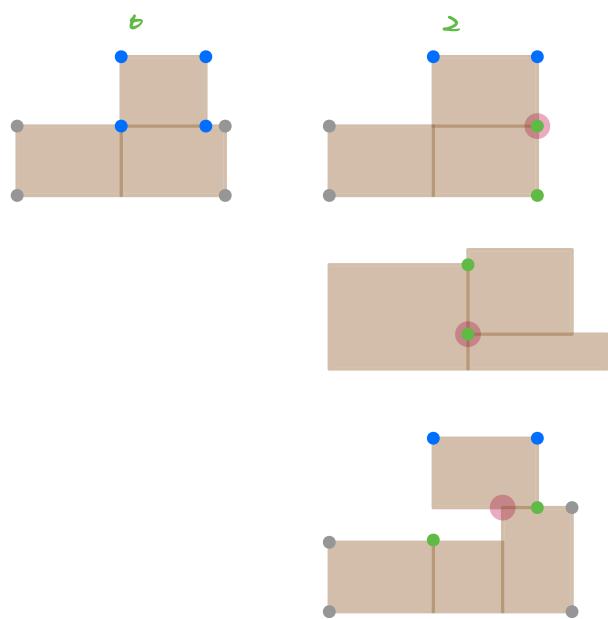
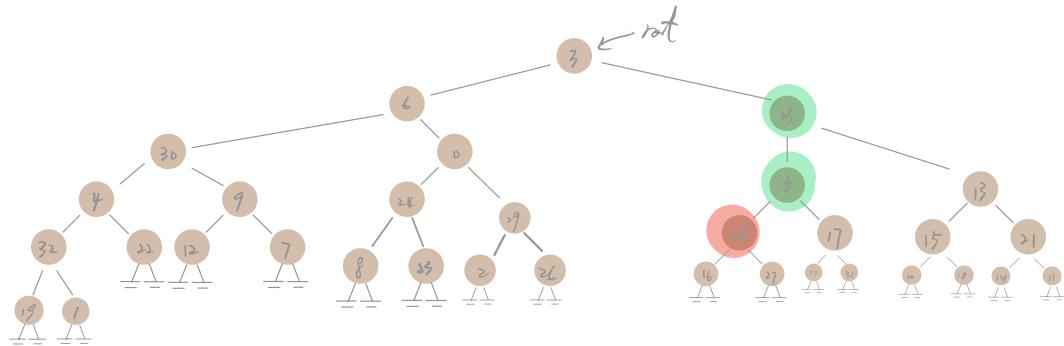


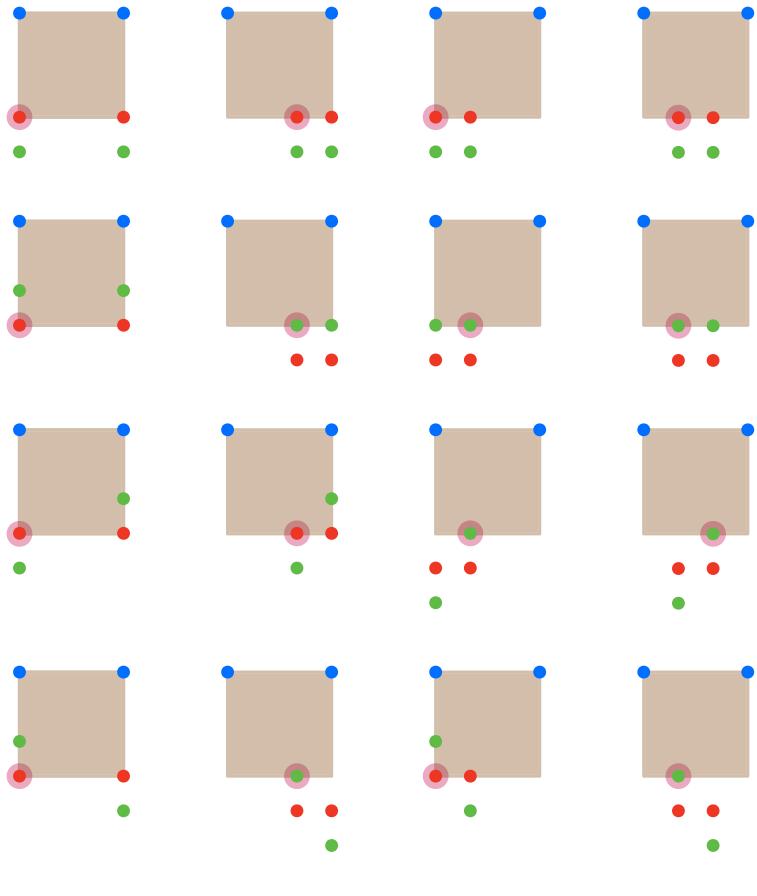
blockchain blockId width height x y lc re pr
bk0 6 336 133 49 497 24 29 5
bk10s 1 119 378 483 84 33 33 32
bk10s 2 140 161 231 630 33 33 29
bk1c 3 349 119 0 6 33 33 29
bk1l 4 175 159 308 0 32 22 30
bk1z 5 406 149 133 847 28 17 25
bk1z 6 140 497 49 0 30 0 3
bk14a 7 119 186 289 33 33 9
bk14b 8 119 294 615 315 33 33 24
bk14c 9 119 161 189 119 12 7 30
bk15a 10 266 119 427 987 33 33 17
bk15b 11 336 119 0 1554 33 33 21
bk16 12 119 126 308 252 33 33 9
bk17a 13 374 182 0 1239 15 21 25
bk17b 14 184 263 560 315 33 33 21
bk19 15 209 119 0 1554 33 33 13
bk19a 16 119 119 0 1554 33 33 9
bk19b 17 284 137 0 1554 33 33 13
bk20 18 350 152 371 938 33 33 15
bk21 19 140 315 602 0 33 33 32
bk3 20 133 315 574 1106 33 33 15
bk4 21 560 130 0 1421 14 11 13
bk5a 22 140 130 308 119 33 33 4
bk5b 23 133 175 385 595 33 33 24
bk5c 24 231 133 385 462 8 23 0
bk6 25 133 315 0 847 5 13 3
bk7 26 182 949 749 333 33 29
bk8a ~~27~~ 210 210 ~~5350723~~ 33 33 28
bk8b 28 126 376 ~~5350723~~ 6 27 5
bk9a ~~29~~ 182 119 49 650 2 6 0
bk9b 30 182 119 180 119 9 6
bk10 31 357 119 0 1554 33 33 17
bk10d 33 119 184 83 1120 11 11 1



九

插入操作
- 样例操作 -





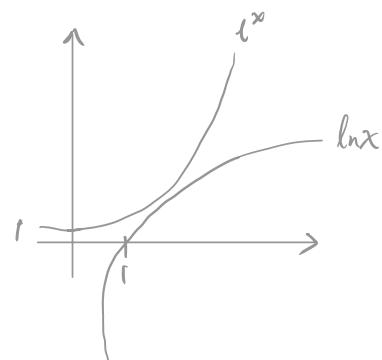
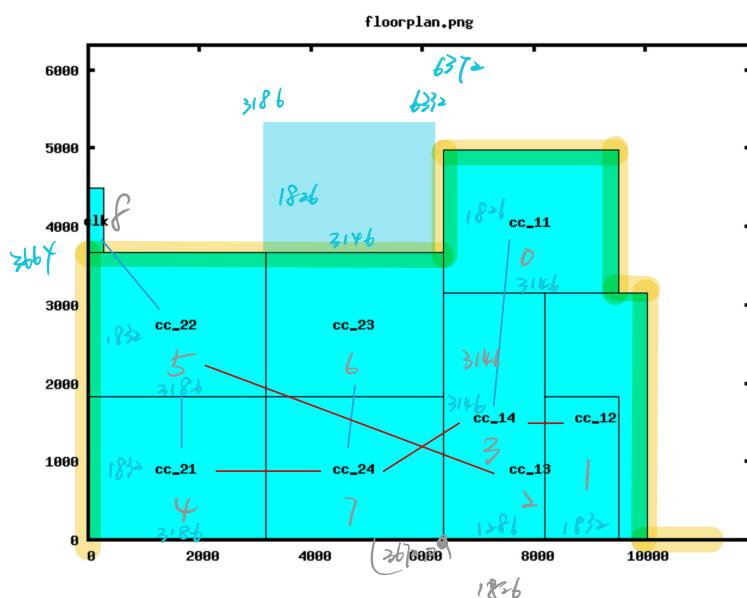
① $y_1 = l$ 上第二 \rightarrow 均最高

② 改 y_2 , 上 $\frac{1}{2}$

③ ~~均~~ 红,

④ 换 y_3

⑤ 若次高量 y_4 ~~均~~
否则 加 y_5



$$\begin{array}{r}
 \cancel{3186} \\
 \cancel{1826} \\
 \cancel{5497} \\
 \hline
 \end{array}
 \quad
 \begin{array}{r}
 \cancel{3186} \\
 \cancel{3186} \\
 \cancel{6372} \\
 \cancel{1826} \\
 \hline
 \end{array}
 \quad
 \begin{array}{r}
 \cancel{6372} \\
 \cancel{3146} \\
 \cancel{9518} \\
 \cancel{7832} \\
 \hline
 \end{array}
 \quad
 \begin{array}{r}
 \cancel{3146} \\
 \hline
 \end{array}$$

$$\begin{array}{r} 178 \\ 1826 \\ \hline 1024 \end{array}$$

$$\begin{array}{r} 1826 \\ 4972 \\ \hline \end{array}$$

$$\begin{array}{r} 1826 \\ 3186 \\ \hline 4978 - \\ \hline \end{array}$$

$$\begin{array}{r} 3186 \\ 3186 \\ \hline 6332 \end{array}$$