

题目：Floorplaning 算法

问题描述：

在经过芯片划分后，整个芯片被分为若干个 Block。在简单情况下，这些 Blocks 均可视为宽高比固定的可移动矩形。芯片上同样摆放着若干 Terminal。不同 Block 之间，Block 与 Terminal 之间会存在相互连接关系，称之为 Net。

衡量芯片 Floorplan 的质量优劣往往采用面积（Area）与线长（Wirelength）两个指标。芯片的面积（**A**）为所有 Block 接后的图形的上、下、左、右边界围成的面积，即能够包裹住 Floorplan 后所有 Block 的最小矩形面积。而芯片的线长（**W**）为所有 Net 的半周长线长之和，即

$$W = \sum_{n_i \in N} HPWL(n_i)$$

其中，N 为所有网络的集合， n_i 为 N 中的一个网络。

最终以加权求和的形式来计算芯片的总 Cost：

$$Cost = \alpha \frac{A}{A_{norm}} + (1 - \alpha) \frac{W}{W_{norm}}$$

其中 α 为面积所占的权重，应在 0~1 范围内。 A_{norm} 、 W_{norm} 分别为归一化面积与归一化线长。为简化计算，不妨令 A_{norm} 为所有 Block 面积之和；而 W_{norm} 为所有网络中的每个 Block 平均边长之和。

题目要求：

1. 遵循给定的输入输出格式，使用 C/C++、python 或 matlab 中一或多种语言编程实现一个简易 Floorplanner，算法不限。
2. 功能要求：
 - a) 根据输入文件.block 中的所有 macro 信息，将所有 block 均摆放在给定

的 Outline 范围内, 且不允许重叠;

- b) 在满足功能要求 a) 的基础上, 根据输入文件.block 与.net, 计算芯片的 Cost, 并使得芯片 Cost 最小;
- c) 在满足功能要求 b) 的基础上, 设计一种方法能够尽可能使得每个网络中相邻的 blocks 更多, 且相邻的边长更长。

- 3. 功能要求中的 a), b)为必做内容, 功能要求 c) 为选做内容。
- 4. 在报告中需说明程序运行方法与项目目录结构, 在文件读写与脚本文件中使用相对路径, 不要使用绝对路径。

输入输出说明:

- 输入:

输入文件有两个, 分别为.block 与.net 文件。其输入格式与说明如下:

- .block

```
Outline: <outline width, outline height>  
NumBlocks: <# of blocks>  
NumTerminals: <# of terminals>  
  
<macro name> <macro width> <macro height>  
... More macros  
  
<terminal name> terminal <terminal x coordinate> <terminal y coordinate>  
... More terminals
```

输入文件 Block 包含有芯片尺寸要求与所有 Block 与 Terminal 的输入信息。Outline 表示芯片的边界信息, 最终 floorplan 后的结果不得超出 Outline 范围。Block, 即上图中的 macro, 输入信息包含其名称、宽度与高度。Terminal 的输入信息包括其名称与坐标。

■ .net

```
NumNets: <# of nets>
NetDegree: <# of terminals in this net>
<terminal name>
... More terminal names
<macro name>
... More macros
... More "NetDegree" and "terminal name"
```

输入文件 Net 包含有芯片内所有互连关系的要求。一个 Net 中可能包含若干个 Blocks 和 Terminals。

● 输出：

输出文件为 .output 文件，其格式与说明如下：

■ .output

```
<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
```

输出包含六个部分：最终的 Cost，总半周长线长 HPWL，芯片面积，芯片的宽与高，程序运行时间（秒），以及模块摆放信息。

■ 可视化

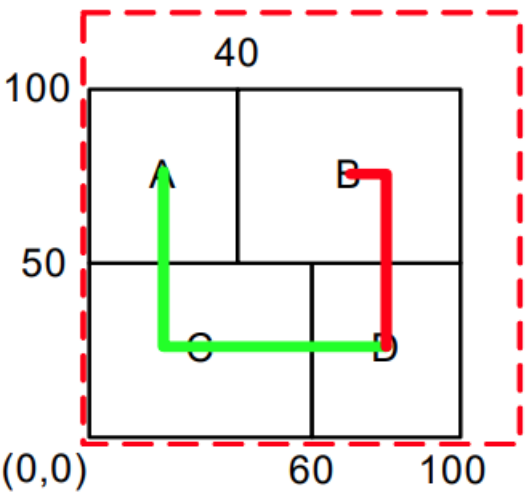
可视化仅为方便调试与完成报告的工具，不计入考核内容。可直接使用或参

照实例程序 draw.py 完成布图结果可视化。

eg. 输入输出文件与 floorplan 结果示例

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)