Algorithms

Programming Assignment #3

Net Routing

選用和 2018 Problem E 相近的讀入方式（分開吃 .pin , .blockage, .net）
執行指令：./Routing <input_file_pin> <input_file_block> <input_file_net> <output_file_name>

3.1 Describe your algorithm to complete the *Routing problem*.

I chose Lee Algorithm (BFS) from EDA textbooks regarding routing in physical design, since its complexity is related to the size of the layout, this should be the most efficient algorithm as the layout size would not exceed 1000*1000. Moreover, the improvements on Lee Algorithm cannot be implemented in C++, it's not practical to use revisions of Lee such as Hadlock's, Soukup's, Mikami-Tabuchi's, Hightower's.

The original version for Lee Algorithm is to guarantee two pins in net would always be connected in the least distance. In this case, I modified it for multiple pins in a net by assuming all pins are wires at first, so that I'd only have to link every wire in sight.

In addition, I used the HPWL (Half-Perimeter Wire Length) method to predict wire-length (according to EDA textbooks) and sort the nets, i.e. choosing net-ordering over rerouting.

Reference: http://users.eecs.northwestern.edu/~haizhou/357/lec6.pdf

3.2 Please analyze the complexity of your algorithm in terms of the number of pins (*p*), the number of blockages (*b*) and the number of netlists (*n*).

The complexity of the algorithm would also be affected by the size of the layout square, with w as its length.

Therefore, the complexity for Lee Algorithm in this case should be O(w*w*(b+n+p)). However, n must be smaller than p, so we can simplify the complexity to O(w*w*(b+p)).