

PA3 Write-Up Json

The main program is "SteinerTreeCreator.cpp" and "SteinerTreeCreator.h". This program implements Prim's algorithm to construct a minimum spanning tree, using the straight distance between two nodes as the weights of the edges. After the tree has been constructed, if the line is neither horizontal nor vertical, it will be unpacked into one vertical and one horizontal line, then dumped into the solution.

The data folder consists of input data files and output result files. The program can be compiled using command:

"g++ SteinerTreeCreator.cpp -o a.out -O3"

then call the executable using:

"/a.out data/case1.txt data/output1.txt"

The cout folder consists of timing and wirelength results for each test case.

"draw.py" is an utility python script to visualize the solution, it can be called by:

"python3 draw.py data/case2 data/output2.txt", and similar to other cases.

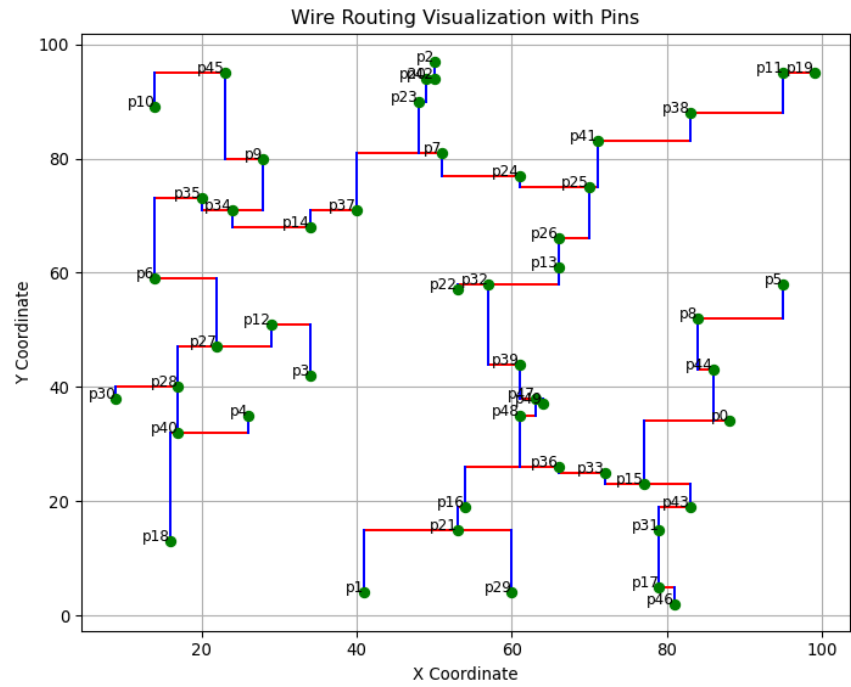
The timing and wirelength results are summarized into the following tables:

Test Case	Wirelength	Runtime [s]
case1	297	3.727e-06
case2	574	3.6722e-05
case3	870	0.00012961
case4	18696	0.00189035
case5	3.34562e+07	0.00519326
case6	1.31216e+09	0.0246207
case7	3.34827e+10	0.151792
case8	6.67965e+11	0.6092
case100000	6.67668e+12	60.3252
case200000	1.33218e+13	251.176
case500000	566228	1005.91

A graph showing a wire routing path (red line) connecting ten pins (green dots) on a grid. The X-axis is labeled 'X Coordinate' and ranges from 0 to 100. The Y-axis is labeled 'Y Coordinate' and ranges from 0 to 100. The pins are labeled p0 through p9. The routing path starts at p5 (10, 73), goes to p2 (10, 37), then to p1 (21, 29), p4 (55, 89), p3 (80, 79), p8 (57, 61), p7 (73, 61), p0 (95, 60), p6 (63, 9), and finally to p9 (15, 16). The path is a single continuous red line connecting all pins in a specific sequence.

Pin Label	X Coordinate	Y Coordinate
p0	95	60
p1	21	29
p2	10	37
p3	80	79
p4	55	89
p5	10	73
p6	63	9
p7	73	61
p8	57	61
p9	15	16

case2:



case3:

