

1. 王竑智 109062542

2. --How to Compile

In HW5/src, enter the following command:

```
$ make
```

It will generate the executable file "hw5" in "HW5/bin"

If you want to remove it, please enter the following command:

```
$ make clean
```

--How to Run

In HW5/src, enter the following command:

```
$ ../bin/<exe> <nets file> <result file>
```

e.g. \$../bin/hw5 *.modified.txt *.result

notice: * means any file name

```
$ ../bin/hw5 ../testcase/ibm01.modified.txt ../output/ibm01.result
```

In "HW5/bin/", enter the following command:

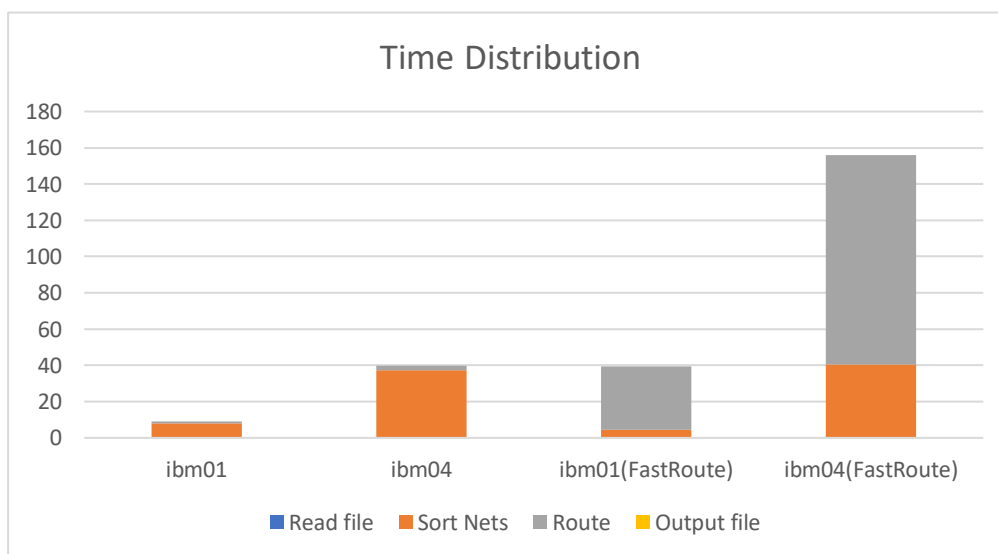
```
$ ./<exe> <nets file> <result file>
```

e.g. \$./hw5 *.modified.txt *.result

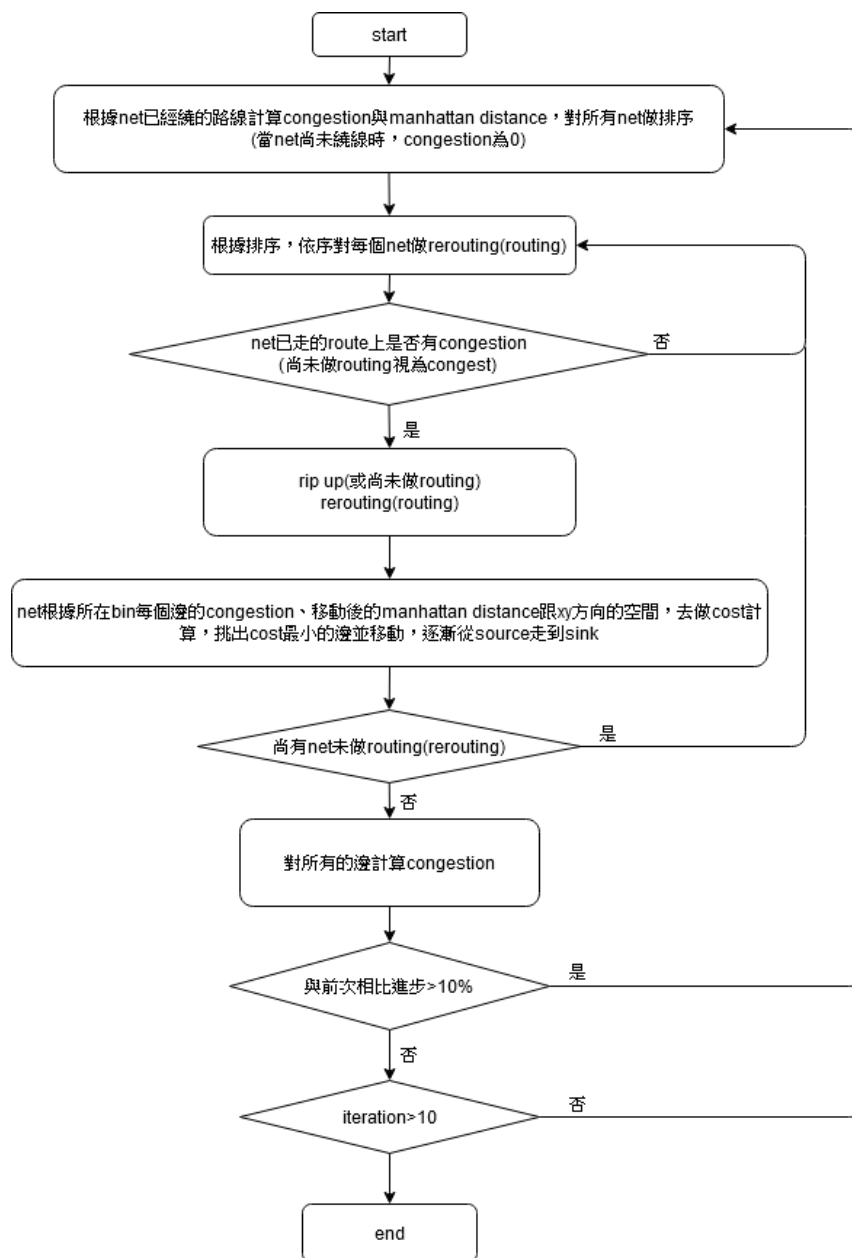
```
$ ./hw5 ../testcase/ibm01.modified.txt ../output/ibm01.result
```

3. Result

Testcase	Total overflow	Total wirelength	Total time(s)
ibm01	2468	56803	8.81
ibm04	4139	154258	39.78
ibm01(FastRoute)	528	60771	39.41
Ibm04(FastRoute)	2407	156926	156.35



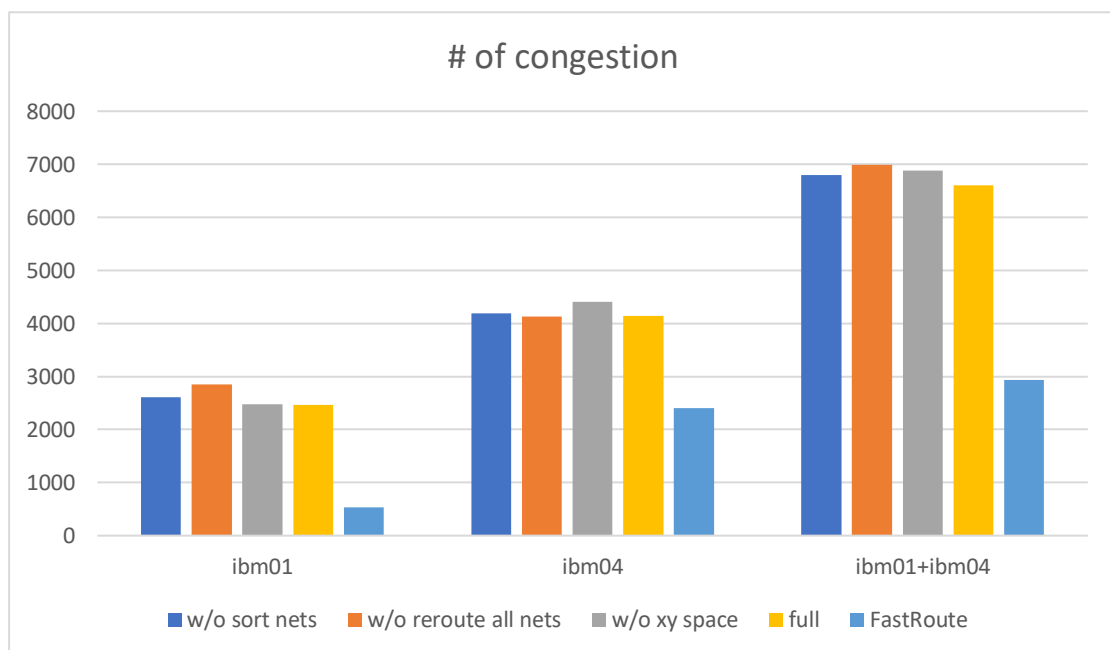
4. Algorithm



我還有用<FastRoute: A Step to Integrate Global Routing into Placement>去做，基本上就是把我的演算法的 cost 改成用 maze routing 去找。

5. 我的驗算法本身不花時間，大部分時間都在 sorting 上，所以我主要專注在如何改進結果，我主要做的有對所有 net 依照 congestion 加上 manhattan distance 做排序，由大到小，因為 manhattan distance 大的 net，使用的邊數較多，所以將 manhattan distance 加 congestion 大的先 reroute，可以避免小 manhattan distance 的 net 為了避開原本就被 manhattan distance 大的 net 所造成的 congestion，做 detour 拉長繞線，並增加使用的邊數，並且 manhattan distance 長的 net 本身，因為原本的距離就長，在不增加 routing 長度的情況下，可以避開 congestion 嚴重的區域，調整 congestion 和

manhattan 相加的比例也可以改善結果，接著是我原本在每個 iteration 都把所有 net 都做 reroute，但因為前面對所有 net 做排序，我發現不該 reroute 所有的 net，因為我前面已經把 congestion 加 manhattan distance 大的 net 都做 reroute 了，也就是有機會使得短 manhattan distance 的 route 變得沒有 congestion，若是再 reroute 可能破壞前面 congestion 加 manhattan distance 大的 net reroute 出來的結果，反而使得結果變差，最後則是不斷調整計算 cost 時，裡面所用到的元素的比例(bin 每個邊的 congestion, manhattan distance, xy 方向的空間)，其中 xy 方向的空間也可以算是改善結果的方法，因為如果 x 方向的 bin 數較多，代表空間較大，那在做 detour 時，沿著 x 方向做 detour 代表可以減少 congestion 的情況，我用了 FastRoute 去做，結果改善蠻多的，但好像我還沒調整到 FastRoute 可以做到的最好結果。



6. 好的 routing 結果其實不好做，因為除了合理的使用資源以外，如何有效的利用可用的空間，或是適當使用，使得所有的 net 都可以找出一個繞線結果，即便完全限制使得沒有 congestion，也未必能使每個 net 都找出一個 routing 結果。