

HW2 補繳

1. 姓名: 黃柏惟
學號: 110164508

2.

```
--How to Compile
in this directory, enter the following command:
$make
It will generate the executable file "hw2" in "HW2/bin/"

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

--How to Run
In "HW2/bin/", enter the following command:
Usage: ./<exe> <net> <cell> <output file>

e.g.:
$./hw2 ../testcases/p2-1.nets ../testcases/p2-1.cells ../output/p2-1.out
```

3.

B	C	D	E	F	G	H	I	J	K
p2-1 cut size	p2-1 runtime	p2-2 cut size	p2-2 runtime	p2-3 cut size	p2-3 runtime	p2-4 cut size	p2-4 runtime	p2-5 cut size	p2-5 runtime
9	0.04	203	1.21	2190	153.78	45187	275.91	124736	273.55

4.

	IO	FM
T1	0.01	0.03
T2	0.034	1.18
T3	0.75	153
T4	1.41	274
T5	3.997	269

這次補繳的作業時做了紀錄 **gain** 的部分，且設定會跑到 **gain<0** 為止，因此越大的 **case**，其 **FM algorithm** 就會有越明顯的時間增加，故 **T4**與 **T5** 接會跑到最後設定的 270 秒還沒辦法收斂到 **gain<0**。

5.

a.這次這份就完全實作了課本內提到的所有 **FM** 演算法的部分，因此得到的結果相較上次自行設定的結束條件更優。

b.有，我有實作 **bucket list**，其就跟講義提到的一樣，因為這樣管理起來更加方便。

c.每次計算後我會去紀錄 **gain** 值，並在當次所有 **cell** 跑完後去看在第幾個的時候出現最佳的 **gain** 並將我的 **bucket list** 回復到該狀態。

d.這次補繳的部分主要是因為計算時間不列入計分，因此採用課本收斂較慢的方法，但是可以得到較佳的結果。

e.我沒有實作平行化。

6. 我的結果相比之前的同學，還是差了一些些，我想可能是因為我沒有初始化

一開始的分組導致的。

7. 這次作業學到最多的就是 **double link list** 的應用，因為一開始我想說直接用 **array** 做就好，結果連第三個 **case** 都跑不出來，後來才認賠殺出改用課本得方法，雖然實作起來很麻煩，需要考慮到很多的 **case**，但是最終卻可以有效的大幅提升運算速度。