

Homework 2: Two-way Min-cut Partitioning

(1) 109062630 Chieh-Yu, Cheng(鄭傑予)

(2) 在 HW2/src 目錄下

compile :

\$ make

run :

\$../bin/hw2 ../testcases/p2-1.nets ../testcases/p2-1.cells ../output/p2-1.out

(3) (4)

	p2-1	p2-2	p2-3	p2-4	p2-5
final cut size	6	147	2227	41349	123188
runtime	0.02	0.39	10.08	129.93	392.00
T_{IO}	0.01	0.01	0.09	0.15	0.20
$T_{computation}$	0.00	0.38	29.97	129.76	391.79

(5)

- I. 終止條件不同，課堂上的 FM 終止條件為 $\max \text{ partial sum} \leq 0$
我的終止條件為 $\max \text{ partial sum} \leq 0$ 或再跑一次 iteration 時間 > 10 分鐘
- II. 有實作兩個 bucket list，A, B 各一個
- III. `int maxPartialSum;` // 記錄最大的 partial sum
`int bestPass;` // 記錄 maxPartialSum 出現在第幾個 pass
`vector<pair<string, bool>> move;` // 記錄每個 pass 的 move
FM 執行結束後，對 initial partition 進行前 bestPass 的 move
- IV.

	Runtime(s)					Cutsizes				
Rank	p2-1	p2-2	p2-3	p2-4	p2-5	p2-1	p2-2	p2-3	p2-4	p2-5
1	0.001	0.09	8.7	3.13	126.12	5	118	717	39633	124079
2	0.001	0.03	3	8.93	13.16	5	208	1028	41962	122888
3	0.01	0.12	4.17	9.6	27	17	126	1389	44127	125270
4	0.001	0.13	2.49	5.89	20.75	6	272	1449	42950	124661
5	0.001	0.09	2.25	11.01	7.56	6	221	1630	46323	125272

我以 cutsizes 為優化目標，造成多次 iteration 有較長的 runtime

- V.
 - 1.若一條 net 左右皆有 locked cell，lock 此 net，加速 update gain
 - 2.initial partition 採用兩種方式來分堆，第一種為依照 net，一次將某 net 上所有 cell 放到同一邊，直到 balance，第二種為依照 cell 由大到小排序，持續搬動直到 balance
 - 3.若 bucket list 中 A 及 B 的 max gain 相同，依 balance 取向選擇，也就是挑選 size 較小那個 bucket 的 cell
- VI. 更熟悉 FM 演算法及 C++ STL 的使用，一開始沒有建立 class，造成程式效能不佳，需多此走訪 vector 及 map，重新規劃 data structure 後大幅優化，因此學習到事前規劃的重要性