

(1) Your name and student ID

Name: 陳俊元 ID: 110065536

(2) How to compile and execute your program, and give an execution example.

進入到/src 目錄中輸入 make 就會在/bin 中產生執行檔，接著在當前目錄輸入
../bin/hw2 <net file> <cell file> <output file>即可產生結果到 output file 中

Ex : ../bin/hw2 ../testcases/p2-2.cells ../testcases/p2-2.nets

(3) The final cut size and the runtime of each testcase

	P2-1	P2-2	P2-3	P2-4	P2-5
Final cut size	248	2720	28404	137874	314551
runtime	0.05	2.71	298.49	298.51	297.33

```
[cychen21@ic51 HW2_grading]$ bash HW2_grading.sh
-----
This script is used for PDA HW2 grading.
-----
grading on 110065536:
tar: HW2/src/main.cpp: time stamp 2022-11-04 12:30:55 is 412.986043456 s in the future
tar: HW2/src: time stamp 2022-11-04 12:30:49 is 406.776449921 s in the future
tar: HW2/output: time stamp 2022-11-04 12:31:14 is 431.629609207 s in the future
testcase | cutsize | runtime | status
p2-1 | 248 | 0.14 | success
p2-2 | 2720 | 2.90 | success
p2-3 | 28404 | 298.49 | success
p2-4 | 137874 | 298.51 | success
p2-5 | 314551 | 297.33 | success
```

(4) Runtime = $TIO + T_{computation}$. For each case, please analyze your runtime and find out how much time you spent on I/O and how much time you spent on the computation (FM Algorithm).

	P2-1	P2-2	P2-3	P2-4	P2-5
Input(s)	0.01	0.053	0.579	2.60	4.746
Output(s)	0.038	0.018	0.195	0.36	0.823
Computation(s)	0.003	2.69	293.579	288.567	282.4
Total(s)	0.51	2.716	294.353	291.527	287.969

(5) The details of your implementation containing explanations of the following questions:

I. Where is the difference between your algorithm and FM Algorithm described in class? Are they exactly the same?

整體演算法大致相同，但我沒有使用 critical net 因為我發現對於程式整體的速度沒有太大的提升，而由於我的程式因為沒有寫好的關係，所以速度有點慢，因此我對整體演算法做了一點調整

A. 我每一次只有搬動 gain 值為正的 cell

B. 若是當前 cost 比之前的 best 還更好我就直接將當下的狀態紀錄下來

```
if(max_partial_sum < partial_sum )
{
    max_partial_sum = partial_sum;
    result_A = A;
```

C. 我只有實作一個 bucket list 下面會由詳細的說明

D. 我在 ic51 的伺服器上使用 random seed 來做 initial partition，我發現前幾個 case 有使用 random seed 可以讓 cut size 更低，特別是前兩個 case 有明顯的改善，我猜測是我的演算法尚有可以改善的空間

E. 由於我的程式無法在助教規定的 300s 內跑完，所以我在 FM process 的時候，每 5 萬個 cells 給 IO 以及 computation 1 秒的時間做計算，以 p2-5 來說因為有 40 萬個 cells，因此程式會在 285 秒的時候強制停止並且輸出成績

```
if(move_cell != 0 && elapsed.count() * 1e-9<runtime)
```

```
else if(total_cell_size==200000)//p2-4
{
    srand(1);
    runtime -= total_cell_size/50000;
}
else if(total_cell_size==400000)//p2-5
{
    srand(1);
    runtime -= total_cell_size/50000;
}
```

```
int runtime = 295; //setting for timeout cases
```

而我為了避免隱藏測資可能跑不完，自己使用了 scripts 產生出 1M cells nets with 20 pin degree，發現 IO 要花 24s 左右，避免最後 timeout 於是我每 20k 個 cells 給 program 1 秒的時間去跑，希望可以避免隱藏測資 time out，舉例:如果測資有 1M 個 cells， $\text{runtime} = 294 - 1\text{M}/2\text{k} \Rightarrow 294 - 50 = 244$ FM 會卡在第 244 秒然後 timeout 去寫出結果

```
[cychen21@ic51 src]$ ../bin/hw2 ../testcases/test_1M.cells ../testcases/test_1M.nets ../output/1M.outs
Input Time measured: 21.726seconds
after pass : 1 fm's cutsize : 950210maximum partial sum : 0
Computation Time measured: 20.8783seconds
Output Time measured: 2.76172seconds
```

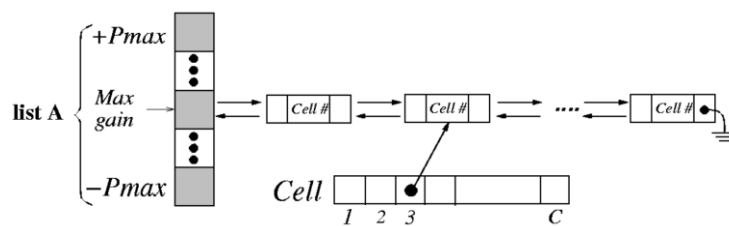
```
else //other cases
{
    runtime -= total_cell_size/20000;
}
```

II. Did you implement the bucket list data structure?

✓ If so, is it exactly the same as that described in the slide? How many are they?

✓ If not, why? You had a better data structure? Or, is bucket list useless?

我有實作 bucket list(以下為講義的 bucket list)



而我的 bucket list 使用以下資料結構，key 一樣為 gain 值而後面使用 unordered_map 而不是講義提到的 link list

```
unordered_map<int, unordered_set<int>> bucketlist;
```

特別是當我實作兩個 bucket list 的時候，效果反而比使用一個還差

grading on 110065536:

testcase	cutsizes	runtime	status
p2-1	299	0.14	success
p2-2	3103	2.07	success
p2-3	30784	202.24	success
p2-4	136255	298.95	success
p2-5	312277	299.75	success

Successfully generate grades to HW2_grade.csv

grading on 110065536:

testcase	cutsizes	runtime	status
p2-1	278	0.11	success
p2-2	2812	2.30	success
p2-3	27929	217.79	success
p2-4	132723	291.26	success
p2-5	312484	292.73	success

使用兩個 bucketlist

使用一個 bucketlist

III. How did you find the maximum partial sum and restore the result?

```
partial_sum += CellGain[move_cell];
UpdateGain( move_cell, Total_A_area,
if(max_partial_sum < partial_sum )
{
    max_partial_sum = partial_sum;
    result_A = A;
```

如(5)B，我根據把每一次的 gain 值都加起來，並且記錄到哪一個階段最佳，但我是直接紀錄最佳 gain 值時候 set 的情況，再第 n+1 次 pass 的時候當作新的 parition

IV. What else did you do to enhance your solution quality (you are required to implement at least one method to enhance your solution quality) and to speed up your program?

因為程式結構不佳的關係，想了很多方法來加速自己的程式，以下是我使用的方式，但我覺得都沒有太大的改變，只有第四項差異最大

1. 使用兩個 bucket list
2. 使用 critical net
3. 在計算 cut cost 時，只要 net 的兩端出現 A/B 就直接 cut cost++ 並且 break
4. 每次計算 cutsizes / balance factor 不應該重算，而是用當下的值是加減(每次重算的話，程式會慢 10 倍)

V. If you implement parallelization (for FM algorithm itself), please describe the implementation details and provide some experimental results.

並沒有實作平行化

(6) What have you learned from this homework? What problem(s) have you encountered in this homework?

由於這是第一次寫 C++ 作業，非常多的程式邏輯有問題，感謝助教有耐心地給予我非常多實用的意見，而以下我也想列出我程式最大問題的部分

1. 應該使用 `struct` 而不是一堆 `unordered_map`，我所有的資訊都是放在 `unordered_map` 裡面，經過助教的提醒我應該把相關的 `cells` `nets` 的資訊使用 `struct`. Ex: `Struct Cell{ int gain, int cell_name }` 等等 然後在用 `vector` 或是 `link list` 把它們串起來。

```
22 // 1-3. A/B set initial partition
23 unordered_set<int> A;
24 unordered_set<int> B;
```

2. 程式執行的時間太慢，導致想了很多方式，但最終才發現是整個結構都有問題，而不是小部分有問題。

然而我擔心無法在 `deadline` 前將程式整份打掉重寫，因此便上傳此份版本，寫下份作業之前我會先把資料結構想清楚，以往常常在寫作業的時候老師並無特別規定時間，只要都跑得出來都還在可以接受的範圍，但上 PDA 的課到目前為止也理解到，在這個領域不止答案的品質重要，程式執行的速度也會大大影響整份 `code` 的水準。

我已經可以預期此次作業的分數應該不會太理想，但我也從連續兩個禮拜的 `coding` 中學到非常多關於 C++ 的知識，包含怎麼使用 `hash_map` / `vector` / `parser` 等等等，除了 `Partition` 本身的演算法以外，也讓我在 C++ 這個工具上有一定程度的認識。

PS. 我為了確認是否是程式問題，把時間拉長到 1000seconds 發現 `cut size` 跟助教的 `baseline` 越來越接近，也 100% 確定我在大的 `case` `cut size` 太高是因為程式跑不完。

```
g++ -std=c++14 main.cpp -O -I ../bin/hw2
nthucad:~/PDA/HW2/src> ../bin/hw2 ../testcases/p2-4.cells ../testcases/p2-4.nets ../c
Time measured: 1000.01seconds
after pass : 1 fm's cutsize : 105118maximum partial sum : 60511
Time measured: 1000.43seconds
after pass : 2 fm's cutsize : 105118maximum partial sum : 0
```

程式太慢，導致 `cost` 無法降低，