

(1) Your name and student ID

111062652 吳仲晏

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

在 HW2/src 中輸入 `../bin/hw2 [cells 檔位置] [nets 檔位置] [output 儲存位置]`

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

Testcases	Cut size	Exe. Time
P2-1	280	0.07s
P2-2	2913	2.77s
P2-3	29583	295.5s
P2-4	136386	297.85s
P2-5	322620	298.68s

(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).

Testcases	TIO	Tcomputation
P2-1	0.02s	0.07s
P2-2	0.1s	2.77s
P2-3	0.97s	295.5s
P2-4	2.43s	297.85s
P2-5	3.18s	298.68s

(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?

執行步驟、理念大致相同。1.挑選最大 gain 值時，若 bucket 內沒有 $\geq 0$ 的 gain 時，就退出不再執行了，雖然有可能沒辦法達到最佳解，可是在時間上來說節省很多(FM 會全部搬動完)。2.實作內使用的資料結構與課程所提及的部分不同。Updating gain 時，使用到 distribution 的幫忙(看 pseudo code 這部分好像比較沒提到)。

II. Did you implement the bucket list data structure?  $\checkmark$  If so, is it exactly the same as that described in the slide? How many are they?  $\checkmark$  If not, why? You had a better data structure? Or, is bucket list useless?

使用 map 做 bucket，格式為`<int,vector<string>>`前面存 gain 值、後面存該格 gain 值有哪些 cells。

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

從 bucket 中 gain 值最大的開始 search&move，直到沒有  $\text{gain} \geq 0$  的 cells。

- 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?

Bucket 內沒有  $\geq 0$  的 gain 值時即停止，因為我的 time complexity 不如助教們的版本來的好，所以雖然 cutsizes 結果不好，但在時間方面可以好一些。

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

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

更加了解 FM 演算法的操作過程及實作時相關資料結構的使用，也讀了一些平行化相關的理論、實作，儘管最後沒能使用平行化實現演算法，過程還是收穫不少。之前沒有碰過 makefile，也在這次的作業學習到 makefile，了解到 linux/unix 中常常使用 makefile 來進程式的部分更新，十分有用的幫助。

Updating gain 的更新用了許久才正確；另外在挑選 base cell 時，感覺可以更加優化時間，但沒想到是哪部分導致時間上的效能沒有很好。

過程中寄信詢問助教，助教也都很有耐心地回復，非常感謝助教的幫忙!!