

CS6135 PDA HW2 REPORT

(1) Your name and student ID

姓名：葉財豪

學號：112062638

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

How to Compile:

```
$ cd /HW2/src/
```

```
$ make
```

```
[u112062638@ic21 src]$ pwd
/users/course/2023F/cs613500110003/u112062638/HW2/HW2/src
[u112062638@ic21 src]$ make
g++ -O3 -std=c++11 FM.cpp main.cpp -o ../bin/hw2
g++ -O3 -std=c++11 FM_parallel.cpp -o ../bin/hw2_parallel
```

Execute:

```
$ ./HW2/bin/hw2 <Input file> <output file>
```

```
[u112062638@ic21 src]$ ../bin/hw2 ../testcase/public1.txt ../output/public1.out
```

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

	public1	public2	public3	public4	public5	public6
Cut size	183	1899	6634	1820	1479	11280
Runtime	0.02	0.33	2.48	0.14	1.13	11.24

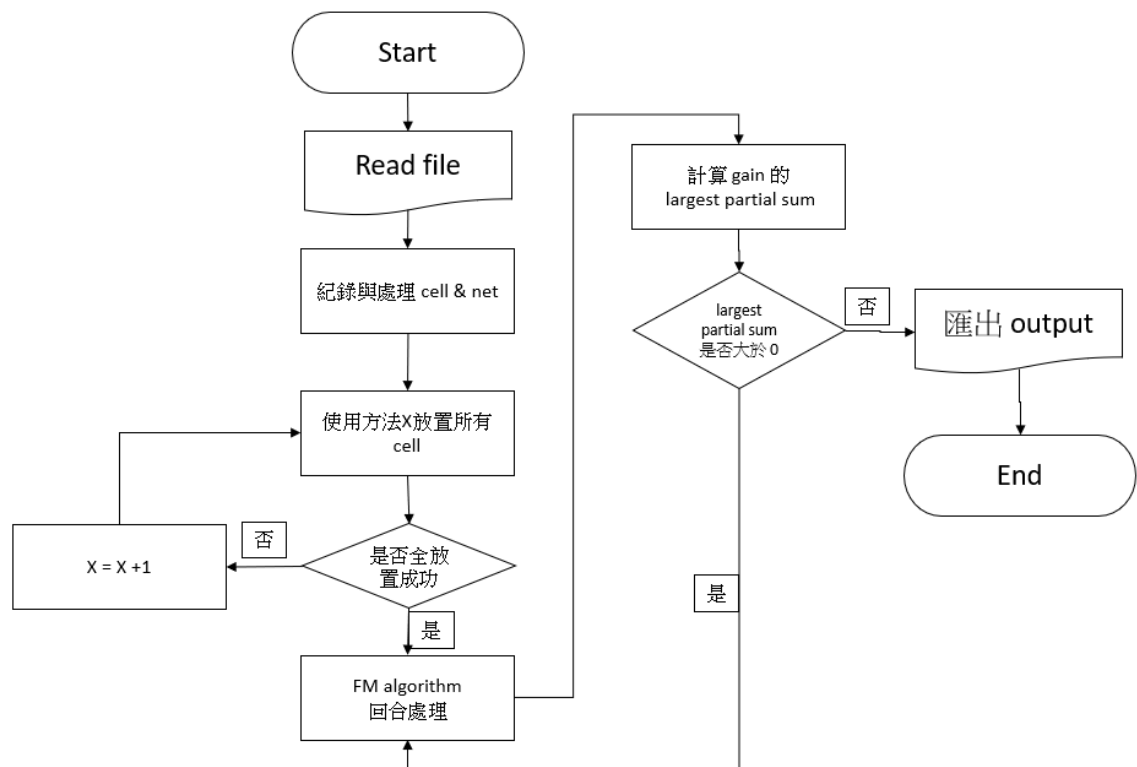
使用 HW2_grading.sh 進行計算

testcase	cutsizesize	runtime	status
public1	183	0.02	success
public2	1899	0.33	success
public3	6634	2.48	success
public4	1820	0.14	success
public5	1479	1.13	success
public6	11280	11.24	success

(4) The details of your algorithm. You could use flow chart(s) and/or pseudo code to help elaborate your algorithm. If your algorithm is similar to some previous work, please cite the corresponding paper(s) and reveal your difference(s).

使用 FM algorithm 進行修改

Fiduccia and Mattheyses. "A linear time heuristic for improving network partitions," 19th Design Automation Conf.,1982.



不同的地方在於 **balance** 的條件，原版為

$$\frac{|A|}{|A| + |B|} \approx r$$

$$rW - S_{max} \leq |A| \leq rW + S_{max},$$

$$\text{When } W = |A| + |B|; S_{max} = \max_i s(i).$$

為了滿足題目的要求改為

$$S_A \leq Area_A, S_B \leq Area_B$$

S_X 是 X 集合的 cell 面積總和

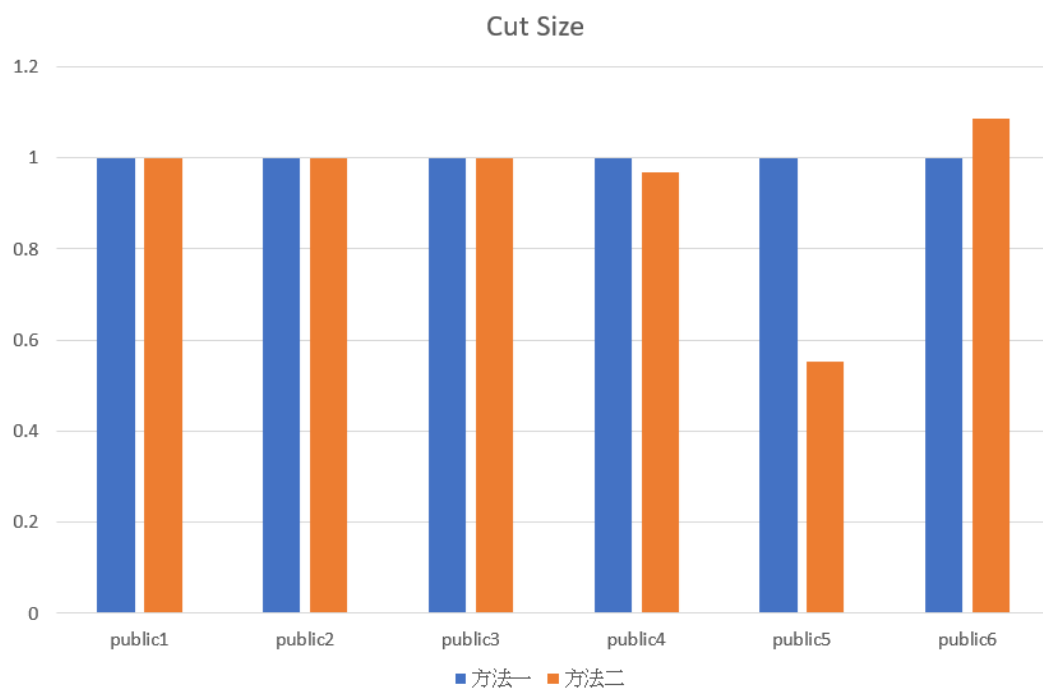
$Area_X$ 是 Die X 可以使用的最大面積

(5) Try your best to enhance your solution quality. What tricks did you use to enhance your solution quality? Also plot the effects of those different settings like the ones shown below

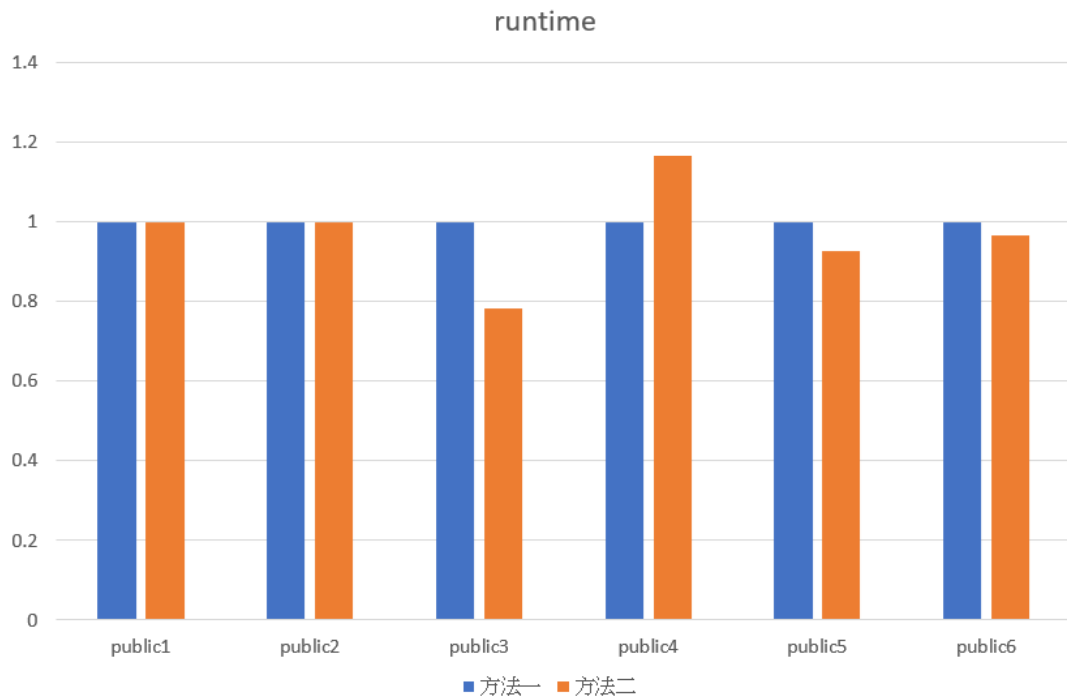
(方法一)初始 cell 放置直接照著輸入的 cell 順序擺放，比較每個 cell 的 $\min(\text{putDieASize}, \text{putDieBSize})$ ，盡量將 cell 放在需要較小面積的 Die 上。

(方法二)先取出所有會大於 $\text{DieSize} \times 0.01$ 的巨型 cell 進行排序，由大到小先放置，因為巨型 cell 的數量不會超過 200 個，所以 sorting 花 constant time。最後再依照 1 的方法放置其餘 cell。

有些輸入當中會出現面積特別大的 cell，一開始使用方法一在這些測資會出現較大的 cut size，所以方法二先處理這些大 cell，在執行時間差不多的情況下，public5 少了 55.4% 的 cut size，public6 則多了 8.6% 的 cut size。時間的部分兩者相差不多。



(數值為 (方法 X/方法一))



(數值為 (方法 X/方法一))

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

沒有實作

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

我學到了盡量紀錄所有已經算過且可能會在被用到的資料，可以大幅降低執行時間、提高可讀性與實作的複雜度，在開發過程當中不是很肯定未來會使用到哪些資料，所以把資料都存起來，在實作過程中就會發現前面已經算過了可以直接拿來使用，方便許多。

許多放置方法沒辦法把所有的 cell 都擺進去，在過程中常是各式各樣的組合，最終版本為組合先前試過的各種方法，在執行時一個個嘗試能不能擺放成功，如果不行就不斷換下一種方式。

在實作 FM 的過程當中遇到的 bug 很難發現錯誤在哪裡，我的解決方式為使用小型測資當作輸入，且在每個 function 稍微檢查輸入與過程是否發生錯誤，一發生錯誤會記錄一些錯誤訊息並直接結束程式，完成一個 function 會花一些時間測試一下。

