

(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/hw4 <aux file><output file>即可產生結果到 output file 中

Ex : ../bin/hw4 ../testcases/ibm01.aux ../output/ibm01.result

(3) The total displacement, the maximum displacement and the runtime of each testcase. Notice that the runtime contains I/O, constructing data structures, computing parts, etc. (You need to

```
grading on 110065536:
testcase | max disp. | total disp. | runtime | status
adaptec1 | 271.97 | 3781517.00 | 0.91 | Maximum displacement constraint was violated for adaptec1.
adaptec3 | 261.40 | 6462287.50 | 2.28 | Maximum displacement constraint was violated for adaptec3.
ibm01 | 2217.52 | 5835513.50 | 0.04 | success
ibm07 | 4620.06 | 30055620.00 | 0.15 | success
ibm09 | 4696.26 | 44921240.00 | 0.17 | success

-----
Successfully generate grades to HW4_grade.csv
-----
```

provide a screenshot for the result of HW4\_grading.sh.)

(4) The details of your implementation. If there is anything different between your implementation and the algorithm in the ISPD-08 paper, please reveal the difference(s) and explain the reasons

大致上是相同，以下條列式初與 paper 不同的部分

1. Paper 中的 weight 是使用 cell 的 degree

Property	Explanation
$x'(i), y'(i)$	Position in global placement
$x(i), y(i)$	Position in legal placement
$w(i)$	Width
$e(i)$	Weight (e.g., number of pins)

但由於我們沒有 cell 的 pins，因此我使用 cell 的 width 來當作 weight

有趣的是使用不同的參數作為 weight 會大大的影響成績

以下為使用 weight \* height 作為成績的分數

grading on 110065536:				
testcase	max disp.	total disp.	runtime	status
adaptec1	337.98	3744777.00	0.82	Maximum displacement constraint was violated for adaptec1.
adaptec3	193.54	6443588.00	2.29	Maximum displacement constraint was violated for adaptec3.
ibm01	8483.19	9681011.00	0.04	Maximum displacement constraint was violated for ibm01.
ibm07	15997.06	78308608.00	0.15	Maximum displacement constraint was violated for ibm07.
ibm09	22414.42	160925056.00	0.17	Maximum displacement constraint was violated for ibm09.

可以看到 ibm 的 testcase 全部都違反 max displacement. 我最後選用 width 當作 cell 的 weight

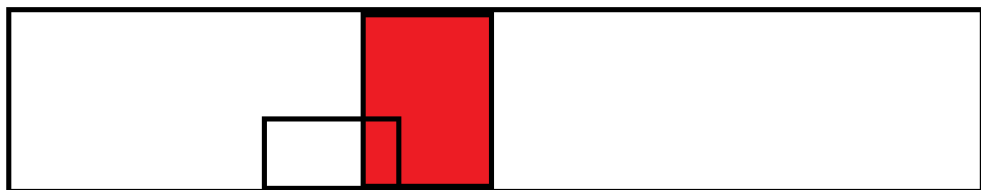
```
Cell(string &name, int &width, int &height)
: name(name), width(width), height(height), weight(width)
```

2.

```
4 | foreach row  $r$  do
5 | |   Insert cell  $i$  into row  $r$ ;
6 | |   PlaceRow  $r$  (trial);
```

第二點比較是對於 coding 的實作不同，我不確定這是否是影響我 adaptec test 都沒有辦法達成 max displacement 的原因，原 paper 提到對於每個 row 都去執行 PlaceRow(trial) 但我覺得這在 coding 上有點麻煩，於是我實作了一個 FakePlace 的 function 來實作 paper 的這個部分，功能為將 cell 擺入對應的 row(subrow)再去看看對應的 cost，過程中沒有真的去建立 cluster 等等，而是在後面 PlaceRow(原 paper 為 place row final)中再去實作，

3. 此次 testcase 有出現 subrow，雖然說 subrow 也可以看成某一個 Row，但是在 Cell 卡在中間時



要選擇把 cell 放在左邊或右邊則為大大的影響答案的品質

```

Total displacement : 4.91055e+06
Max displacement : 1204.28
../testcase/adaptec3
Total displacement : 6.74434e+06
Max displacement : 2084
../testcase/ibm01
Total displacement : 5.8355e+06
Max displacement : 2217.52
../testcase/ibm07
Total displacement : 3.00555e+07
Max displacement : 4620.06
../testcase/ibm09
Total displacement : 4.49211e+07
Max displacement : 4696.26

```

以上可以看到 ibm 沒有影響，因為 ibm test case 沒有 terminal

(5) What tricks did you do to speed up your program or to enhance your solution quality? Also plot the effects of those different settings like the ones shown below

根據上述提到的點(1)

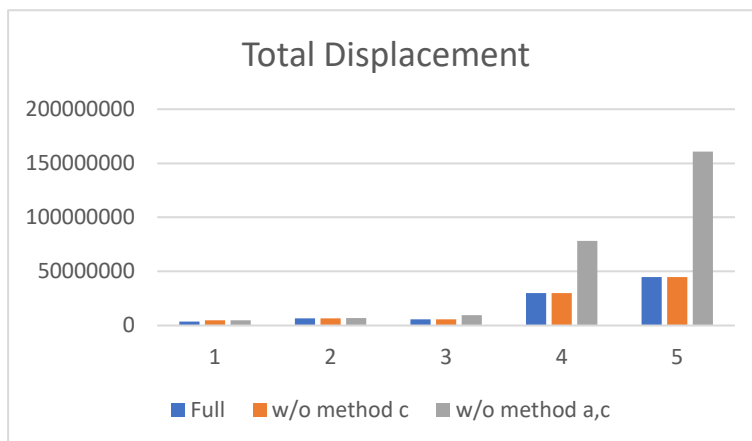
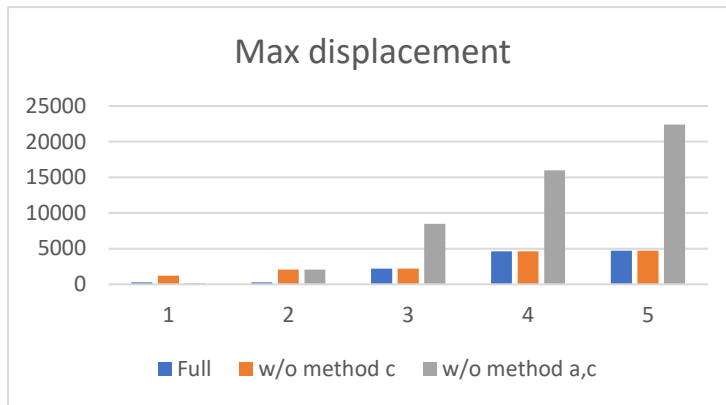
我主要的更動都對於 displacement 有很大的改變，但是對於 runtime 沒有太多差別，因為 paper 即有提到

a. 對於 cell 可以擺入的 row，不需要每個 row 都去看，根據 paper 定義的 cell move vertically，因為如果連 y 方向的 cost 都大於 cost，那就不用再往上(下)找，

all rows of the chip (line 4-10). Rather, each cell is first moved to the nearest row (according to the global position). Then, it is moved above and below this row. For each row, a lower bound of the cost is computed by assuming that the cell is only moved vertically. If the lower bound exceeds the minimal cost of an already found legal position, then the movement in this direction (up or down over the rows) can be stopped. This method limits the movement to a few

b. 找 row 的時候也只需要找離 cell GP 最近的那個 row 去進行擺放，之後再根據那個 row 進行 up/down 的探索即可

c. 同 4-(3)



- (6) Please compare your results with the previous top 5 students' results, and show your advantage either in runtime or in solution quality. Are your results better than theirs?

我覺得我的成績非常接近去年的前五名，大致每個 case 都在他們伯仲之間，run time 上來說我覺得是差不多的，但我的 case 有些情況不太穩定，照理說如果有一些好，應該要全部都超越才對，可能 code 的部分還是有可以改進的地方

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

我沒有實作平行化的 code

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

這是我第一次直接邊看著 paper 邊把 code 寫出來，其中理解到 pseudo code 要變成直接可以執行的 code 有一些困難，前兩次作業老師都對於細節有很深入的講解，但這次更多的部份要靠自己去研讀 paper 的細節，而我上面提到的那幾點，隨便一點就大大的影響 quality，可能只是一兩行 code 會讓整個 displacement 有 10 倍以上的差距，也讓我知道很多細節是 paper 不會特別跟你說清楚，要靠自己實作時去發現的。此次 coding 有幾個非常有趣的 coding 疑問

1. 我在 append vector 的時候，使用 insert / copy 兩個 function 居然會導致結果不一樣，很顯然是有未定義好的部分
2. 我在 parser 的時候，因為 read node 沒有寫好，導致有時候會漏爬一行測資，還讓以為是助教的 grading 有問題，最後發現還是自己的 code 怪怪的，但第一次遇到這種對 binary 執行會有不一樣的 output，對於這邊有甚麼問題，有啟發我的好奇心，但我需要一些時間研究
3. 這次最讓我印象深刻的是 paper 的很多沒有提到的細節我以為沒什麼關係，結果因為那一兩行讓答案差距非常大，讓我知道對於很多細部的條件判斷要再思考的更深入一點

結論：這次不知道為甚麼 adal 3 兩個 case 永遠沒辦法滿足 max displacement，雖然 total displacement 的結果感覺還不錯，應該還有更多的細節沒有思考清楚。