

## Hw3 report

### (1) Your name and student ID

學生姓名:黃柏惟

學號:110164508

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

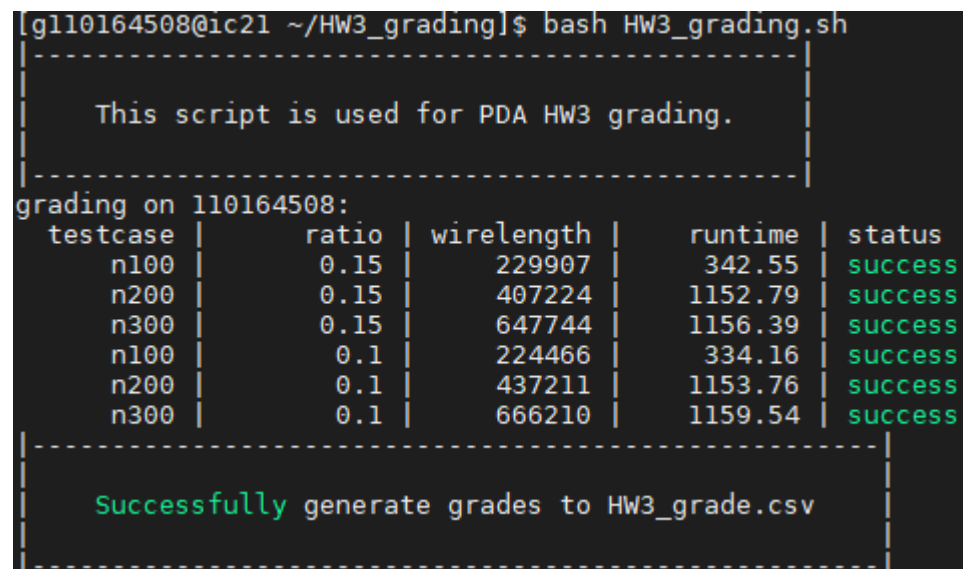
```
--How to Compile
  in this directory, enter the following command:
  $make
  It will generate the executable file "hw3" in "HW3/bin/"

  If you want to remove it, please enter the following command:
  $make clean

--How to Run
  In "HW3/bin/", enter the following command:
  Usage: ./<exe> <hardblocks> <nets> <pl> <output file> <dead space ratio>

  e.g.:
  $./hw3 ../testcases/n100.hardblocks ../testcases/n100.nets ../testcases/n100.pl ../output/n100.floorplan 0.1
```

(3) The wirelength and the runtime of each testcase with the dead space ratios 0.1 and 0.15, respectively. Notice that the runtime contains I/O, constructing data structures, initial floorplanning, computing (perturbation) parts, etc. The more details your experiments have, the more clearly you will know where the runtime bottlenecks are. You can plot your results like the one shown below.



```
[g110164508@ic21 ~/HW3_grading]$ bash HW3_grading.sh

-----
This script is used for PDA HW3 grading.
-----

grading on 110164508:
  testcase |      ratio | wirelength |      runtime | status
  -----|-----|-----|-----|-----
  n100     |      0.15 |    229907 |      342.55 | success
  n200     |      0.15 |    407224 |     1152.79 | success
  n300     |      0.15 |    647744 |     1156.39 | success
  n100     |      0.1  |    224466 |      334.16 | success
  n200     |      0.1  |    437211 |     1153.76 | success
  n300     |      0.1  |    666210 |     1159.54 | success
  -----

Successfully generate grades to HW3_grade.csv
```

Case1 約跑 320 秒，其 wirelength 大約 225000

Case2 約跑 1150 秒，其 wirelength 大約 420000

Case3 約跑 1150 秒，其 wirelength 大約 630000

其中遇到的瓶頸是跑進 fixed outline 花費太多的時間，造成沒有太多的時間可以去收斂 wirelength，甚至偶爾也會有失敗的情況發生。

其中因為我是用 class 去儲存因此每次計算都需要複製一份不同地址的 copy，我認為是我程式中最花費時間的部分，不過因為時間問題來不及修改資料結構。

(4) Please show that how small the dead space ratio could be for your program to produce a legal result in 20 minutes.

如果不考慮 wirelength 的話的優劣的話全力去壓面積約可以跑進 0.07 左右的 dead space ratio。

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

我的作法是採用課本教的 B\* tree 的方式建立 tree 然後透過 SA 收斂。

其中在計算 cost 的時候除了課本提及的 area、wirelength 以及 aspect ratio，還另外添加了 x 與 y 方向的 penalty 將超出的部分給予懲罰來協助收斂。

(6) 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.

這次跟課本比較不一樣的地方在我在 cost function 的地方給了超出 fixed outline 的 penalty 期望它能夠更快的收斂進規定的區域內。

還有我調整了不同溫度下的運算次數讓前期在擾動時運算較少，而在後期多運算幾次藉此找到最佳解。

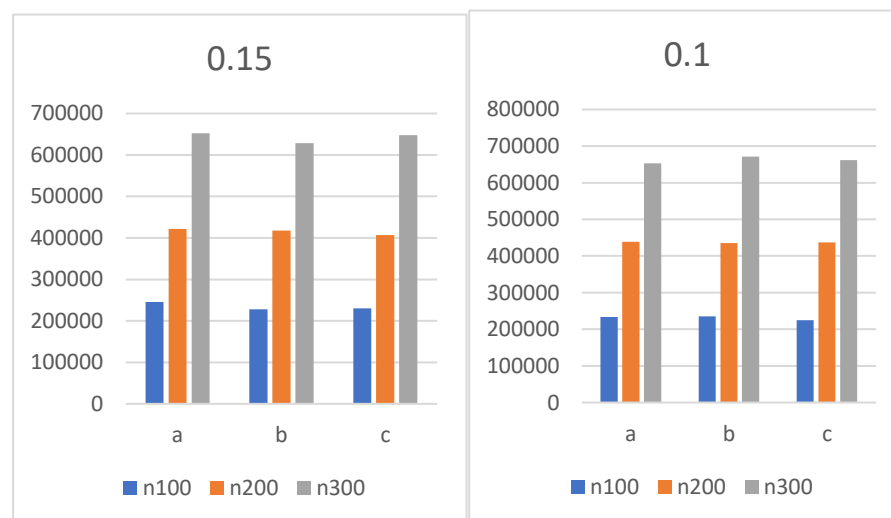
(比較表)

方法 1:  $3 \cdot (\text{area}/\text{init\_area}) + 1 \cdot (\text{wire}/\text{init\_wire}) + 3(1-R)^2 + 10 \cdot x\_penalty + 10 \cdot y\_penalty$

方法 2:  $1 \cdot (\text{area}/\text{init\_area}) + 1 \cdot (\text{wire}/\text{init\_wire}) + 0.6(1-R)^2 + 10 \cdot x\_penalty + 10 \cdot y\_penalty$

方法 3: 方法 2+降低 T 下降速度

方法 3 雖然花最久但是效果相較於另外兩種都比較好，並且由於沒有調整得很好的關係因此每次做出來的結果上下浮動也蠻大的。



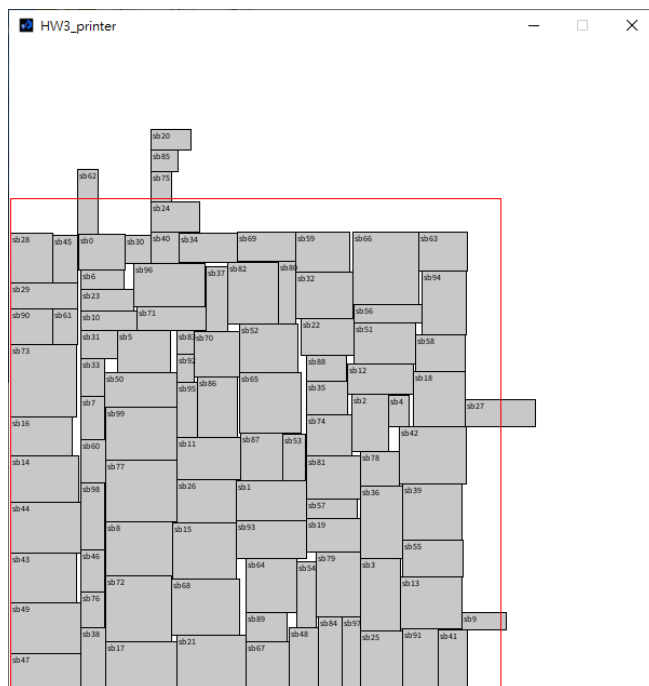
(7) Please compare your results with the top 5 students' results last year for the case where the dead space ratio is set to 0.15, and show your advantage either in runtime or in solution quality. Are your results better than theirs?

我的結果皆比去年 top5 還要差上不少，因為我花大部分時間在想辦法塞進 fixed outline 內，因此 run time 以及 quality 皆比他們還要差上不少。

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

這次算是第一次實作這種需要調整參數的作業，一開始以為把架構架出後很快地就可以做完作業，沒想到卻頻頻發生超出結果的情況發生，因此最後花費很多時間在調整參數的調整上。

我也學習到要把東西塞進一塊決定好的區域內是比想像的複雜的。



若不考慮 fixed outline 的話有測出排得還不錯的數據不過很容易卡在 local min。