

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

Ans:

In HW3/src enter "make" to compile and the command in make is:

```
$ g++ -O3 -std=c++11 main.cpp -o ../bin/hw3
```

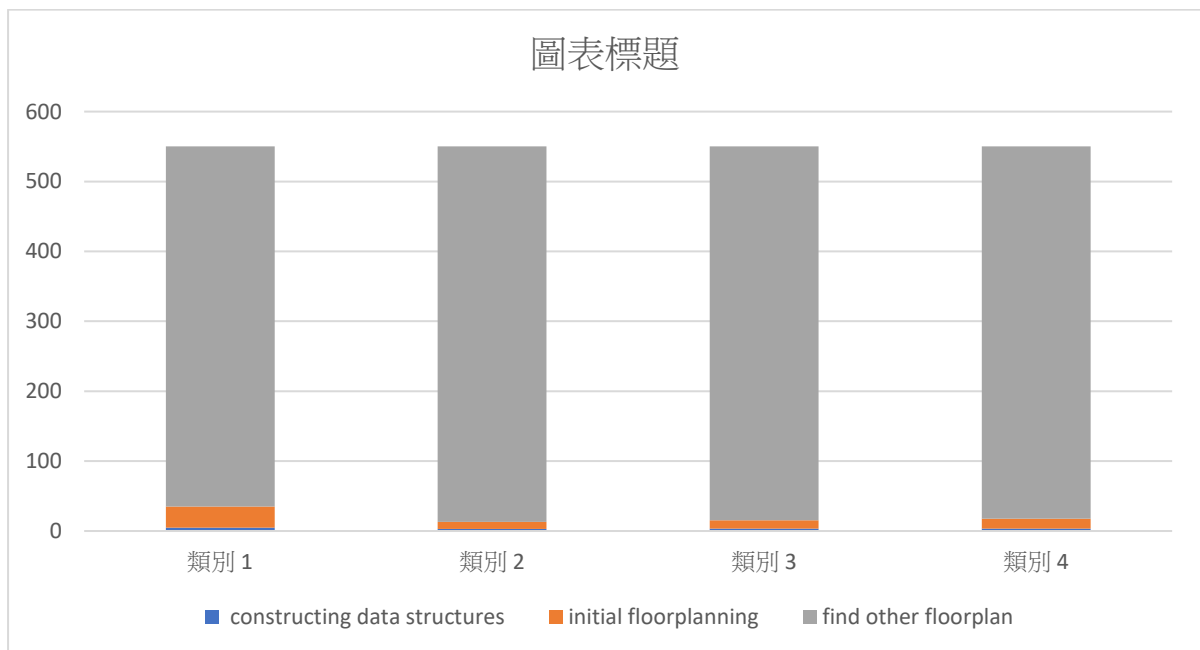
After compile change the directory to HW3/bin, enter the following command: `./hw3 <txt file> <floorplan file>`

For example if we want to test public1.txt then enter the command:

```
$ ./hw2 ../testcase/public1.txt ../output/public1.floorplan
```

(3) The wirelength and the runtime of each testcase

	Public1	Public2	Pulic3	Public4
Total Wirelength	241372396	29818880	3447895	125477200
Run time(s)	550.05	550.0	550.01	550.9



(4) How did you determine the shapes of the soft modules? What are the benefits of your approach?

Ans:我認為 soft module 的 shape 對於線長的影响有限比較多的是影响是前 n

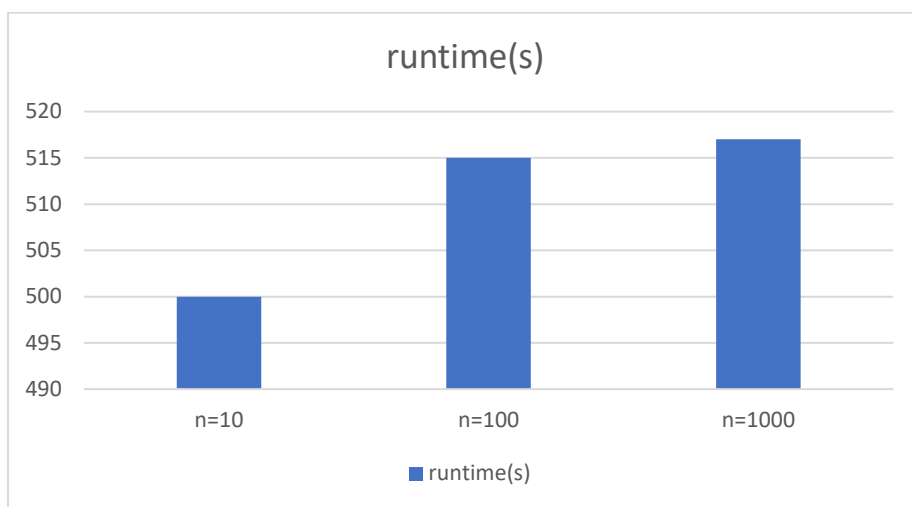
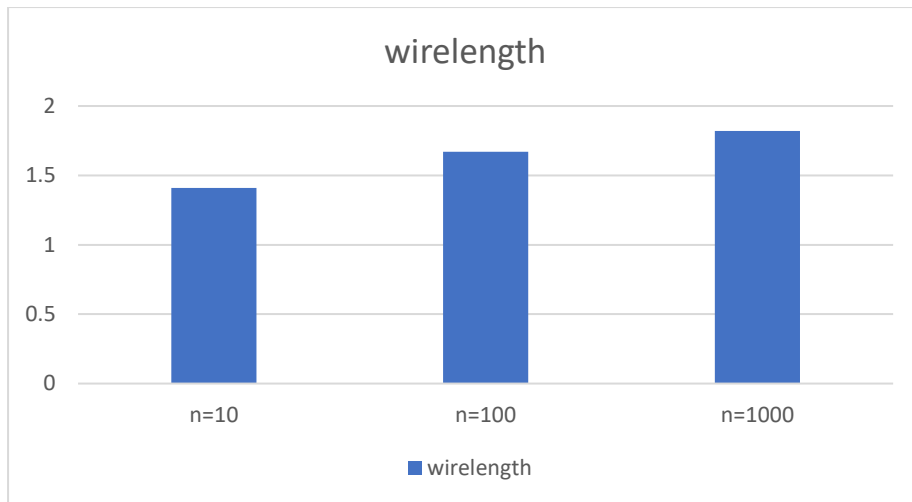
個 soft module 的 shape 會不會導致接下來的 soft module 放不進去。所以我的方法就是去試試看一個 soft module 的所有 w, h 只要沒有 overlap 就直接擺進去。好處就是希望這個 module 用盡所有的合法的 shape 都要擺得進去讓所有細小的空間都能利用到以提高接下來擺整個 floorplan 會合法的機率。

(5) The details of your floorplanning 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).

Ans:假設有 n 個 soft modules 並且將每個 soft modules 用 index $0 \sim n-1$ 表示也就是 $m_0 \sim m_{n-1}$ 。令一個 chip 的寬、高為 w, h 。由於 aspect ratio 必須是 $0.5 \sim 2$ 而且每個 soft module m_i 都有一個 minimum area 所以可以求得所有合法的正整數 w, h 組合並且存到對應到 m_i 的 vector w_i 中。我的作法就是跑過 chip 的每一個點都去試試看某一個 module 能不能擺得進去(沒有和其他 module 重疊)，也就是跑一個 nested for loop 從 $i = 0 \sim i = w$ 並且在每個 i 再從 $j = 0 \sim j = h$ 並且每個點都從 module $0 \sim$ module $n-1$ 的所有 w, h 組合一個一個試，只要能擺得進去的就直接擺進去直到所有 soft modules 都擺進去。由於前面有說到一個一個 module 去試是照著 $m_0 \sim m_{n-1}$ 這個順序去試的接著我就把 $m_0 \sim m_{n-1}$ 的幾種排列(ex: $m_1, m_2, m_3 \Rightarrow m_1, m_3, m_2 \Rightarrow m_2, m_1, m_3 \dots$)都重新去跑跑看前面的方法並且 output 出其中最小的總線長的 floorplan。

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

Ans:在我的方法中我是試各種 module 的排列以及所有 module 的合法長寬組合，但很明顯不可能試的完所有的排列所以只能試其中的幾個，而在擺一個 module 時若去試所有的長寬組合則會嚴重拖慢一個 iteration 的時間導致能試的排列少了，所以我不跑一個 module 的所有合法長寬組合而是指挑其中 n 個去試並且試試看 $n = 10, n = 100, n = 1000$ 。



所以最後選擇了 $n=10$ 。

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

Ans:此次作業中我並沒有實作 parallelization。

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

Ans:我認為這次作業比起上次作業難的地方是更需要注重合法性，在一些 testcase 中很容易亂放就會有 module 重疊的情況，在 floorplan 中有很多限制像是不能重疊、soft modules 的長寬比不能隨意還有不能超過晶片範圍等等所以我學到了除了注重線長下應該要變成先求合法性再求縮小線長的策略。