HW3 Report

111065519 林詰珊

1. How to compile:

step1. 進入資料夾

```
$ cd HW3/src
```

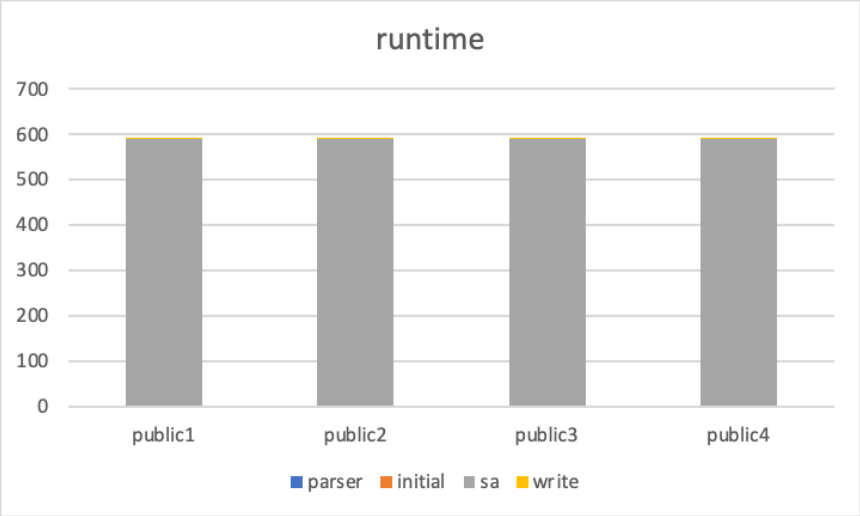
step 2. 執行程式

```
$ make
$ ../bin/HW3 ../testcase/public1.txt ../output/public1.floorplan
格式為(../bin/HW3 ../testcase/<input filename>.txt ../output/<output filename>.floorplan
```

2. The wirelength and the runtime of each testcase

	public1	public2	public3	public4
wirelength	185546910	25744421	3181477	100674825
runtime	590.07	590.06	590.04	590.06

從下圖的runtime分佈可以看到我大部分的時過都是在做SA，而且都是在我設定的時間到達時停止，代表我的方法可能initial的不是非常好，所以他會收斂很久都找不到停下的地方直到超過時限



3. How did you determine the shapes of the soft modules? What are the benefits of your approach?

我使用random的方法得到一個初始的soft modules形狀，同時這個形狀也要符合modules的比例和面積規定，得到初始化的形狀之後，會再繼續經過SA做perturbation，做resize或rotate來去挑選出最合適的soft modules形狀。我認為這樣的方法有以下幾種優點：

1. 多樣性和探索性： 隨機初始化可以產生各種不同的形狀，這有助於探索設計空間並發現可能的解決方案。
2. 防止陷入局部極小值： 如果初始化過於集中在某個區域，可能會陷入局部極小值，所以通過隨機初始化，有機會在空間中的不同位置尋找解決方案，有助於避免陷入極小值。

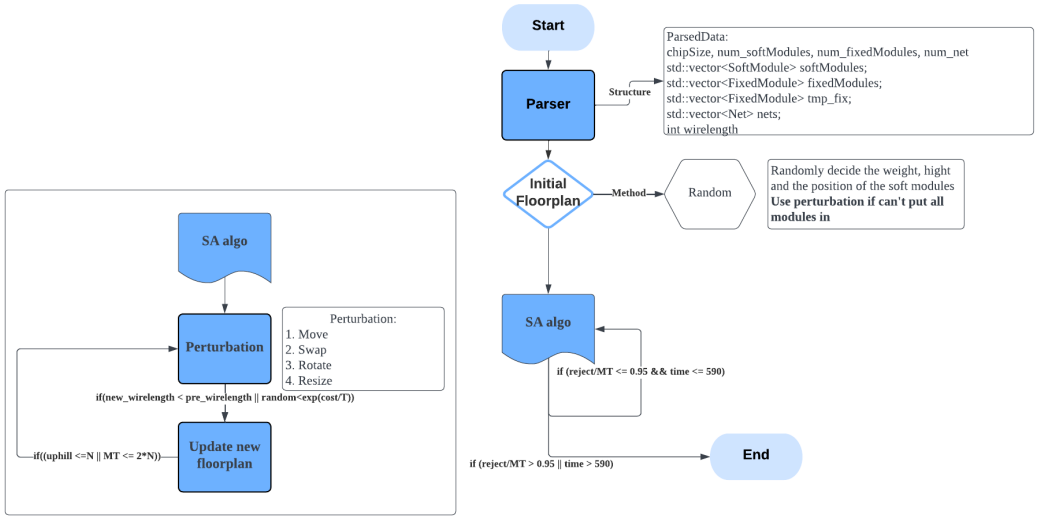
4. Algo detail

- **Parser:** 將.txt中的資訊放到設定好的data structure中，這邊我使用vector來去儲存soft和fix的modules; 同時，這邊會計算出各個soft modules 長寬的最大最小值，並且根據soft modules 的minArea由大到小做排序，以便後面在做initial的時候可以根據面積大小來決定modules的擺放順序
- **Initial Floorplan:** 以前面決定的長寬間距和已知的minArea為條件，random出softmodules的weight, hight和position，每次random完後需要檢查是否會與已經放置好的modules重疊(包括fixmodules)，直到所有softmodules都合法地放進layout中

  Tips: 這邊因為有可能會一直random不到可以放的位置，所以我將SA中使用的perturbation引入到initial階段一起使用，假如random到一定次數都擺不進去的時候，

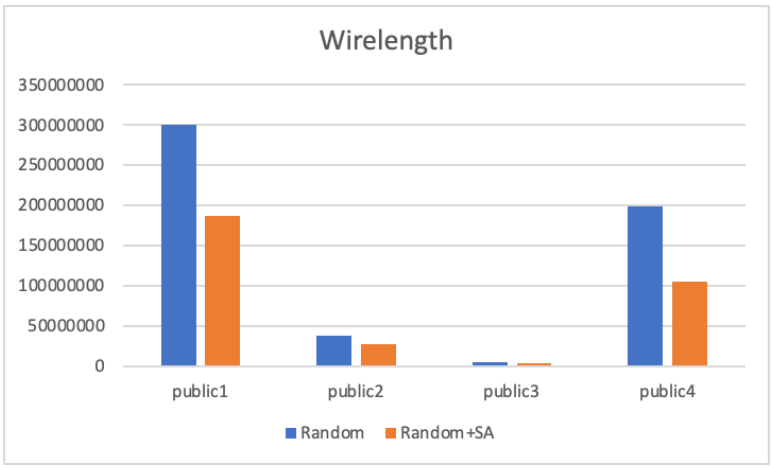
就會對前面已經擺放過的soft modules做隨機一種perturbation(move, swap, rotate, resize)，做完後再重新擺放本次要放入的module

- **SA algo:** SA的部分主要與課本上的一樣，我會對initial的floorplan結果做隨機一種perturbation，若wirelength有比較好我就會接受這個新的floorplan，並為了避免結果卡在local minima，若做完perturbation沒有比較好的話就會根據temperature計算一個機率決定要不要接受這個比較差的結果，而這個部分會一直執行直到拒絕率超過95%或是時間到590秒



5. Solution enhance

這次的作業我比較了random的方法與random+SA的wirelength比較，根據實驗結果，加上SA之後的會讓wirelength大幅減少，這次因為SA通過設定temperature會選擇性地接受一些比較不好的結果來讓wirelength不會卡在local minima，可以繼續往下找到global minima。我認為這次的作業initial也很重要，如果initial的好其實很快就會找到global minima，如果不好的話就要等很久甚至超過規定的運行時間才可以找到，但因為暫時還想不到其他比較合適的方法來做initial，所以我就暫時先使用random的方式做擺放。



6. Parallelization

No.

7. What I learned from this homework?

這次的作業我發現我會沒有思考整個程式的脈絡就開始寫code，導致會遇到一些小bug常常要找很久，像是這次需要設定很多規則來去限制soft modules的擺放，不只是width和height的最大最小值，當modules做perturbation後會不會超過chip outline等等的規定也都要注意，我一開始因為沒有想到這個問題所以在擺放的時候一直出錯，後來是畫出擺放結果發現的，但如果我一開始就有想清楚所有擺放的規則，就不會因為這件事卡很久。另外，因為在寫SA的時候有reject/MT的這個判斷式來檢查reject數量是否有超過規定的機率，但因為一開始的MT是0，若是放在分母就會跑錯，所以我就在這裡意識到do…while()和while()的差異。