

(1) Your name and student ID

吴仲晏 111062652

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

--How to Compile

In HW3/src, enter the following command:

\$ make

It will generate the executable file "hw3" in "HW3/bin/".

If you want to remove it and all .o files, please enter the following command:

\$ make clean

--How to Run

In this directory, enter the following command:

Usage: ../bin/<exe> <hardblocks file> <nets file> <pl file> <output file>
<dead_space_ratio>

e.g.:

\$../bin/hw3 ../testcase/n100.hardblocks ../testcase/n100.nets ../testcase/n100.pl ../
output/n100.floorplan 0.15

(3) The wirelength and the runtime of each testcase with the dead space ratios 0.15 and 0.1, respectively.

File	Dead space ratio	Wirelength	Time	I/O time (read+ write)	Init floorplan time	Construct DS	perturbation
n100	0.15	245045	2.13s	≈0.01s	≈0	≈0.01s	1.58s
n200	0.15	446388	14.48s	≈0.01s	≈0	≈0.01s	6.13s
n300	0.15	676892	24.24s	≈0.01s	≈0	≈0.01s	21.83s
n100	0.1	246121	168.46s	≈0.01s	≈0	≈0.01s	167.04s
n200	0.1	451278	12.28s	≈0.01s	≈0	≈0.01s	10.78s
n300	0.1	688028	21.27s	≈0.01s	≈0	≈0.01s	17.48s

(4) Please show that how small the dead space ratio could be for your program to produce a legal result in 10 minutes

file	Dead space ratio
N100	0.089
N200	0.088

N300	0.07754
------	---------

- (5) The details of your implementation. If there is anything different between your implementation and the algorithm in the DAC-86 paper, please reveal the difference(s) and explain the reasons.

一開始照<高度,寬度>大到小排序，由下往上排列，若滿足 fixed outline 在 perturbation 時也要滿足 fixed outline 才會更新答案。

沒有建 polish expression 的樹，而是使用 vector<string>儲存，並用 stack push 和 pop 的方式對後序進行座標的計算。由於建樹很花時間，所以採用 vector 配合 stack 計算較為快速且方便。

- (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

一開始先排序好後檢查是否符合 fixedoutline 規則，1.若符合:perturbation 試圖找出好的 wirelength。這邊的 perturbation 若 reject 超過 500 次則退出，因為發現若一開始就滿足面積要求，但是不斷 perturbation，雖然 wirelength 會縮減，但是花費的時間是好幾十倍，我認為效益不高。2.不符合:使用 SA 不斷嘗試，cost function 使用到最右邊(block+block.w)的位置與限制範圍的距離平方做判斷。

Perturbation 時有可能越來越往差的方向移動，所以判斷時 reject 會計算 reject 幾次，若超過設定數字則把 polish expression 設為前幾回的最佳結果從新開始 perturbation，試圖不要因為某一次錯誤的 perturbation 就往錯誤的方向進行。但是在 SA 時仍能很難滿足 fixedoutline，要是使用平行化同時多種嘗試可以加快速度。

- (7) Please compare your results with the previous top 3 students' results 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?

Ranks	Wirelength			Runtime (s)		
	n100	n200	n300	n100	n200	n300
1	<u>207309</u>	<u>367785</u>	<u>504903</u>	<u>13.97</u>	<u>84.54</u>	263.33
2	209351	379674	521749	25.57	99.49	<u>209.78</u>
3	210220	392175	544879	37.45	105.83	486.73

Me 245045 451860 676892 1.05s 6.4s 24.24s

沒有，在 n100、n200 的 wirelength 差距較為不明顯，但是到了 n300 的 wirelength 差距就被拉開了。以第一名來說，wirelength 又短、速度又快，可

能是 init floorplan 就擺的蠻好。我可能需要在 initial floorplan 的地方加強。

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

沒有使用平行化。

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

學習到了 wirelength 的計算、stockmeyer algorithm、SA algorithm、polish expression 計算、vector 中 capacity 和 size 的特性。