

# PD PA1 Report

## 1.Data structure used in my program

因為程式內包含許多資料結構，所以我會描述程式內所運行的演算法，並說明各部份所使用的資料結構為何。

首先，我的演算法主要使用上課所教的F-M heuristic，且是建立於作業檔案內所附的範例往下修改與實作，所以我是用class Partitioner作為演算法主架構。以class Net作為紀錄net的資訊。以class Cell紀錄cell的資訊並實作cell的移動，同時對每個cell都建出各自的Node，Node用於建立doubly linked list，用以實作計算及更新gain時的bucket list。

關於parse input，我是直接使用作業範例本身寫好的function，但因為我沒用到netname，所以我拿掉這部分，並去掉\_netName2Id的map。

我的F-M heuristic主要都在partition的function內，而其中又可分為以下幾步驟。

### a. init partition

首先要將cell分在a、b兩partition，我先對每個cell紀錄他的netlist中net size最大的值，再利用該值對cell做sort，以此作為init partition的順序，之後輪流放入a、b partition。

以下是while(cut decrease > 0)會做的事

### b. init bucket lists and variables

這裡我會根據cell跟net的資訊計算每個cell的gain，之後根據cell的gain建立a與b各自的bucket list，bucket list的資料結構為map，map中的key為gain值，value為Node的pointer，以此建立Node的doubly linked list。此處每次插入同gain值的Node(Cell)時，都是從doubly linked list的頭插入，並將map中的value改為新插入的Node的pointer。最後init Partitioner中用來記錄的variable。

### c.main algorithm

當還有可以移動的cell時，我會利用bucket list找出符合balance的max gain cell，取法是從bucket list的gain值最大的doubly linked list的頭取出，也就是Last in First out，之後令其移動partition，lock該cell，計算目前累積gain值，並以兩個vector分別紀錄此步移動的cell與獲得gain值，最後update受影響的cell的gain，再重新檢查是否有可以移動的cell進行迴圈。

同樣，在update cell的gain後要update其在bucket list的位置時，插入新gain值的doubly linked list的方法一樣是Last in First out原則，remove則是doubly linked list的remove。

### d.find largest gain

當cut decrease > 0時(也就是最大累積gain值>0)，我會利用先前提到紀錄移動cell與獲得gain值的兩vector，將其從尾巴不斷還原cell的移動，直到vector中的gain值總和與最大累積gain值相同即停止，則此partition即為一次F-M迴圈後所求partition。

以上是while(cut decrease > 0)會做的事

### e. calculate cutsizes

在輸出前計算出最終cutsizes。

最後是write result, 我是直接使用原本作業範例內的write result function, 並無修改。

## 2.My findings

首先是依照作業範例給定的bucket list資料結構所用來update gain所花費的時間不如我想像的高。因為原先上課所學時, 我認定bucket list是用vector加doubly linked list實作, map的各操作複雜度為 $O(\log n)$ , 所花時間可能要多, 但實作後發現兩者實作bucket list所花時間相差不大, 因此最後選擇使用範例所給的map。

再來是不同init partition的造成的cutsizes結果讓我出乎意料, 我試過用random、依照cell id、依照pin num、依照cell的netlist的netsize總和、以及cell的netlist的max netsize對cell做sort再放入a、b partition, 最後效果(依照evaluator計算)排序大致為  
max netsize > 依照pin num >> random = 依照cell id = netsize總和  
沒想到init partition結果差異那麼大, 有些case cutsizes可以差到接近2倍, 因此最後選擇max netsize作為sort標準。