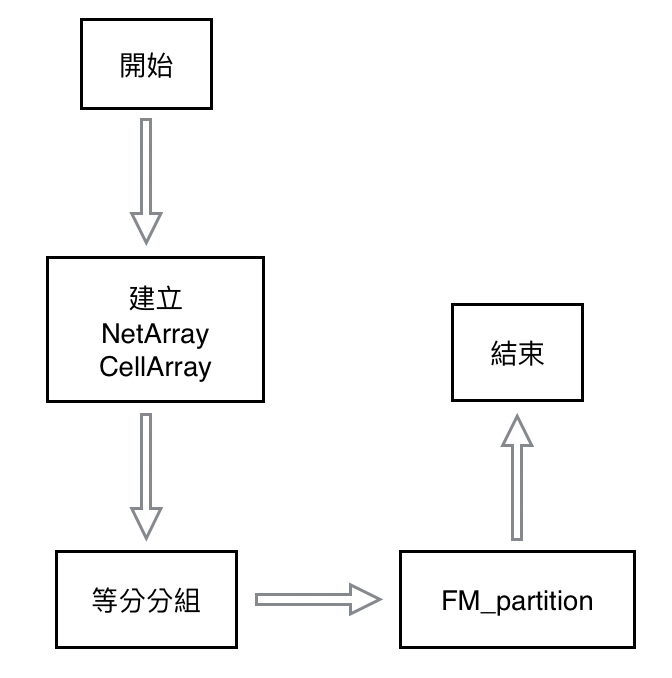
實作 (Implementation)

1. **演算法流程 (Algorithm Flow)**



1. 在讀input檔的時候，會先建立NetArray，之後再利用NetArray資訊建立CellArray，方便之後能夠直接抓到Net或Cell的connecting information。
2. 接著會簡單的將所有Cell等分成GroupA 及 GroupB
3. FM\_partition會不斷的執行，直到沒有更好的partition結果。
4. **Pseudocode**

**Input**: Connecting information of all nets

**Output**: Two groups with minimal cutsize

**begin**

Construct NetArray and CellArray by the input connecting information

Initial GroupA and GroupB

**while**( there exists better partition result with respect to cutsize )

Move cells between GroupA and GroupB

Output the final GroupA, GroupB and the corresponding cutsize.

**end**

1. **資料結構 ( Data Structure )**

(1)

由於在updateCellGain時，會判斷Cell上每個Net的 From Block 及 To Block 的 Cell 個數，因此每個Cell會紀錄自己是否在GroupA，而每條Net 也會紀錄在GroupA 及 GroupB 的Cell 個數。同時，Cell也要記錄自己在gainBucket裡的位置以方便更新gainBucket，以及用來判斷是否locked的flag。每個Cell及Cet也會紀錄自己的id。

Class Cell

{

private:

string \_id; // cell name

bool \_inA; // if this cell is in GroupA

int \_gain; // gain of cell

static unsigned \_globalRef; // locked reference

unsigned \_ref; // if this cell is locked

list< Cell\* >\* \_list; // list in gainBucket that contains this cell

list< Cell\* >::iterator \_it; // position in the corresponding list

};

Class Net

{

private:

string \_id; // net name

unsigned \_sizeInA; // number of cells in GroupA

unsigned \_sizeInB; // number of cells in GroupB

};

Class CellArray

{

private:

unordered\_map< string, NET\* > \_NetsInCell; // nets connecting to each cell

};

Class NetArray

{

private:

unordered\_map< string, CELL\* > \_CellsInNet; // cells connected by each net

};

而整個FM\_partition的資料結構為：

Class FM\_partition

{

private:

double \_balFactor; // balance factor required

NetArray\* \_nets; // connecting info of each net

CellArray\* \_cells; // connecting info of each cell

unordered\_map< string, Net\* > \_nHash; // hash to directly access a net

unordered\_map< string, Cell\* > \_cHash; // hash to directly access a cell

unordered\_map< string, Cell\* > \_grpA; // cells in GroupA

unordered\_map< string, Cell\* > \_grpB; // cells in GroupB

map< int, list< Cell\* >\* > \_gainBucket; // cells sorted by gain

};