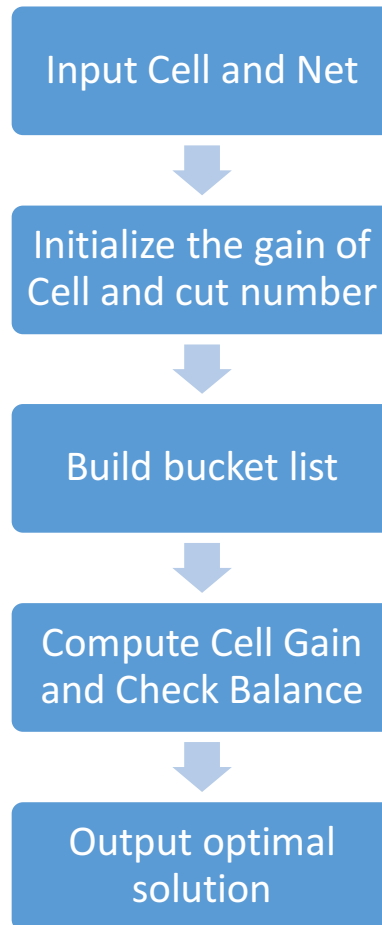


學號：R05943092

姓名：曾育為

實作 (Implementation)

(1) 演算法流程 (Algorithm Flow)



演算法流程圖

配合流程圖，將程式運作流程再加上文字敘述如下：

- (i) 讀入檔案，將 cell 和 net 各自建立資料結構，並將 net 連結所觸及的 cell，net 連結所有觸及的 net
- (ii) 取最大 cell ID 切半，分成左右二堆，並依據左右二邊的 cell 分佈及 net 連線，初始化 cell 的 gain 及 cut 數量
- (iii) 建立一個 bucket list，並將 cell 的 gain 初始化的結果填入 bucket list 中
- (iv) 取最大 gain 並檢查是否維持 balance function，如符合平衡，則將 cell 移到另外一邊，反覆進行 cell 總數的次數則停止
- (v) 在 cell 總數次數中，取出最好的結果，並輸出結果

(2) 虛擬碼 (Pseudocode)

```
Input:  $C = \{C_1, C_2, C_3, \dots, C_n\}$  、  $N = \{n_1, n_2, n_3, \dots, n_m\}$ 
Output: Cutsizes 、 G1 、 G2
begin
Divide all cell into two group. Initialize all the gain value of cell.
Build bucket list to record all the gain value of cell.
for( each cell )
    update all the gain value of cell.
    Check the balance after the cell move
Get the optimal gain value
Output the cut size and left number 、 right number
end
```

(3) 資料結構 (Data Structure)

下面將介紹程式中主要的資料結構：

(i) Cell: 代表輸入檔的 cell

```
class Cell
{
public:
    unsigned        _name; //紀錄 cell 本身的 ID
    string           _side; //紀錄 cell 在左邊 or 右邊
    int              _gain; //紀錄 cell 的 gain 值
    bool             _locked; //紀錄 cell 是否固定
    vector<unsigned> _net_id; //紀錄 cell 連接 net 的 id
};
```

(ii) Net: 代表輸入檔的 net

```
class Net
{
public:
    unsigned        _name; //紀錄 net 本身的 ID
    unsigned         _l_num; //紀錄左邊區域此 net 所連結的 cell 個數
    unsigned         _r_num; //紀錄右邊區域此 net 所連結的 cell 個數
    vector<unsigned> _cell_id; //紀錄 net 連接 cell 的 id
};
```

(4) 實行結果：

	Input_0	Input_1	Input_2	Input_3	Input_4	Input_5
Balance factor	0.2	0.01	0.02	0.1	0.01	0.01
Net number	166998	5000	10000	88888	166998	483599
Cell number	150750	3000	7000	66666	150750	382489
Cut number	14871	1492	2642	31855	54614	170000
Runtime(S)	103.612	0.087	0.3748	62.0489	318.418	3143.55