# 314553017 蕭希敏

## Result

(2) The final cut size and the runtime of each testcase. Paste the screenshot of the result of running the HW2_grading.sh as the picture shown below.

```
nthucad:~/HW2/HW2_grading> bash HW2_grading.sh
+-------------------------------------------------+
|                                                 |
|    This script is used for PDA HW2 grading.     |
|                                                 |
+-------------------------------------------------+
host name: nthucad
compiler version: g++ (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0

grading on 353310918:
In file included from main.cpp:17:
bucket.h: In member function 'void Bucket::erase(int, int, std::__cxx11::list<int>::
iterator)':
bucket.h:31:20: warning: unused parameter 'cell' [-Wunused-parameter]
   31 |     void erase(int cell, int gain, list<int>::iterator it) {
      |                ~~~~^~~~
 checking item          | status
------------------------|--------
 correct tar.gz         | yes
 correct file structure | yes
 have README            | yes
 have Makefile          | yes
 correct make clean     | yes
 correct make           | yes

  testcase |      #ways |   cut size |    runtime | status
-----------|------------|------------|------------|--------
   public1 |          2 |        416 |       0.22 | success
   public1 |          4 |        950 |       0.38 | success
   public2 |          2 |       5398 |      15.90 | success
   public2 |          4 |       7898 |      24.25 | success
+-------------------------------------------------+
|                                                 |
|   Successfully write grades to HW2_grade.csv    |
|                                                 |
+-------------------------------------------------+
nthucad:~/HW2/HW2_grading>
```

## Question

### I. Where is the difference between your algorithm and FM Algorithm

described in class? 一樣

### II. Did you implement the bucket list data structure?

有，但我只使用一個bucket list，因我的class cell有存取自己的group

### III. How did you find the maximum partial sum and restore the result?

我紀錄了移動順序`move_history`並存取`best_step`，restore時從initial state依據move_history還原到best_step，實測下來比複製best step時的state還快

IV. How did you modify the FM algorithm to handle four-way min-cut partitioning? You could use flow chart(s) and/or pseudo code to help elaborate your algorithm.

```
Start
 │
 ▼
讀入資料，建立 Instance
 │
 ▼
以多次 Pass 的 FM 演算法進行第一次 2-way partition
 → 將分為 Group 0 與 Group 1
 │
 ▼
對每個分區再各自進行一次比例限制的 2-way partition：
 ├─ 對 Group 0：
 │     建立子電路，依其佔全體比例 x 計算上下限，
 │     執行 FM_r_optimized，分為 Group 0、Group 2
 │
 └─ 對 Group 1：
       建立子電路，依其佔全體比例 x 計算上下限，
       執行 FM_r_optimized，分為 Group 1、Group 3
 │
 ▼
將子分割結果映射回原電路，合併為四群 {0, 1, 2, 3}
 │
 ▼
輸出最終分割結果
 │
 ▼
End
```

先對整體電路進行一次 2-way FM（Group 0、Group 1），再將 Group 0 分為 Group 0 與 Group 2、Group 1 分為 Group 1 與 Group 3，然後使用改良的FM_r_optimized 它在每次 cell 移動前會進行比例檢查 (feasible_r)，確保符合ratio的限制，基本上邏輯都跟FM一樣。

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

更加熟悉了FM演算法以及將一個方法完整實作出來。實作FM時主要是遇到蠻多bug要de，常常會有小邏輯錯誤導致迴圈無法停止，4 way部分也花了不少時間實作。