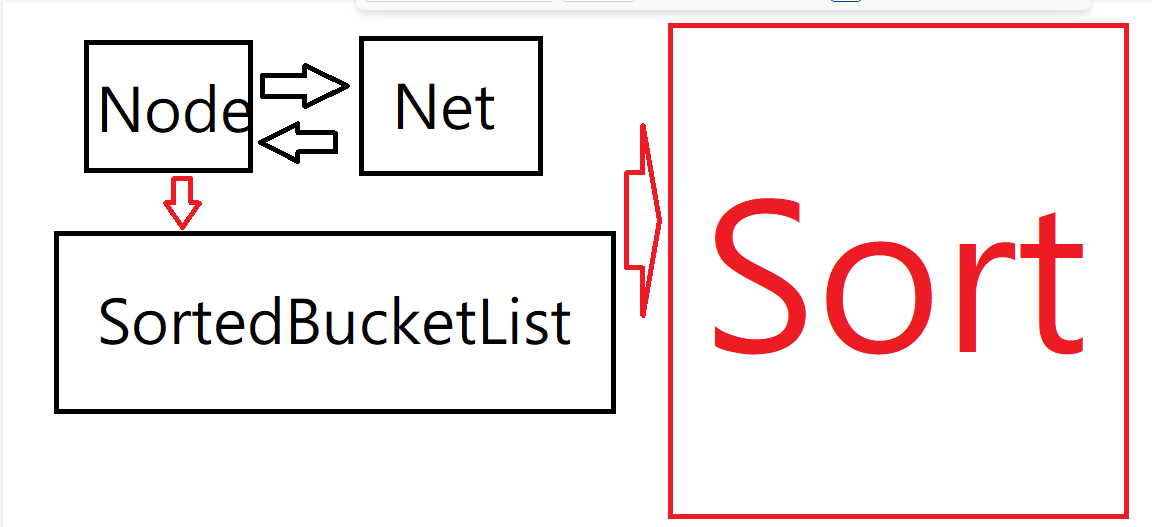
Partition

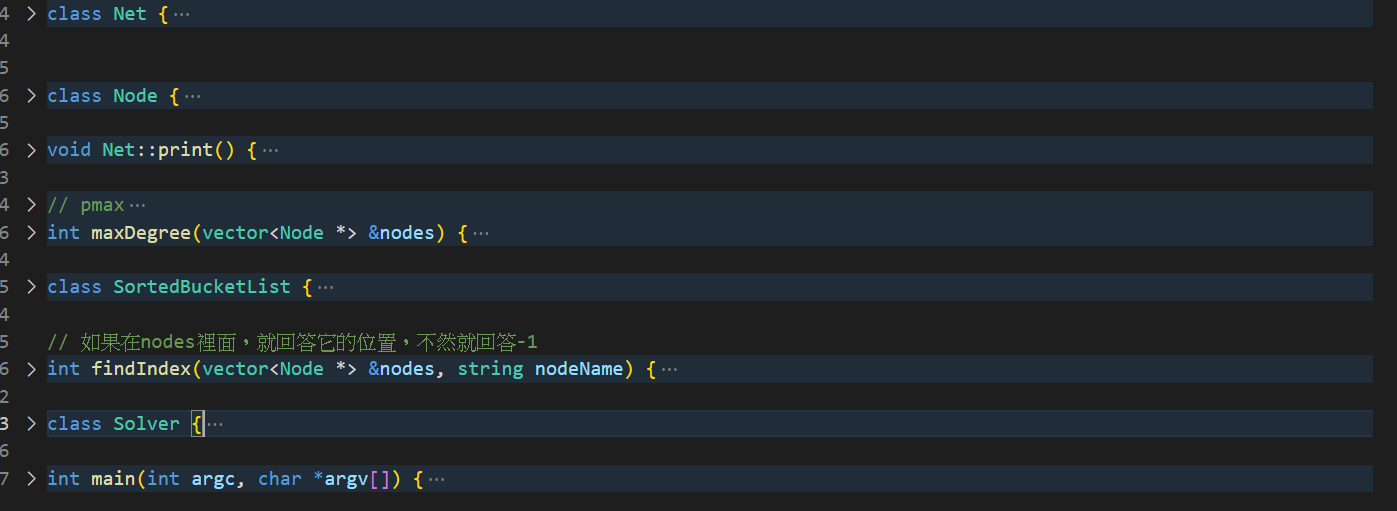
老師: Andy Yu-Guang Chen

姓名:洪嘉均

學號:110521022



(架構圖)

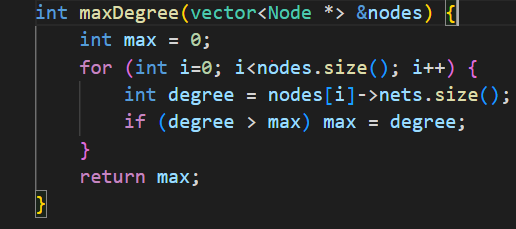


(Code 架構圖)

1. Node

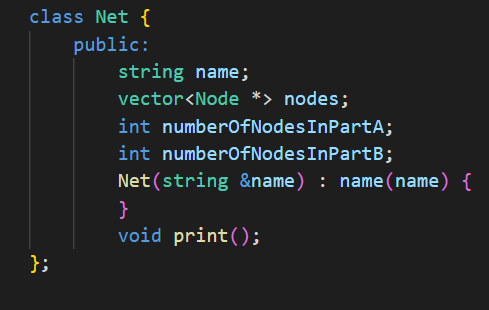


bool isInPartA : 反正只會有兩個地方，一個是PartA，一個是PartB，所以只要我確定我的Node是在PartA就回傳true給我，反之假如是在PartB的話就回傳一個false給我。



pmax = max {pi} = max { ci連到幾個net | 所有node ci }

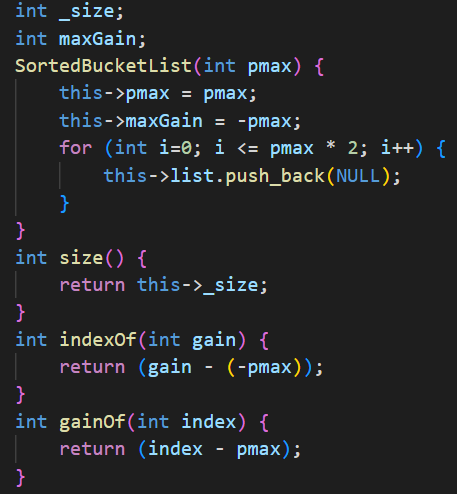
1. Net



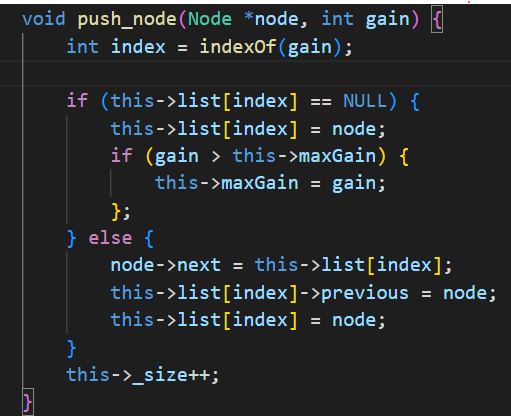
(3) SortedBucketList

int pmax : 要知道pmax，才知道vector要開幾格

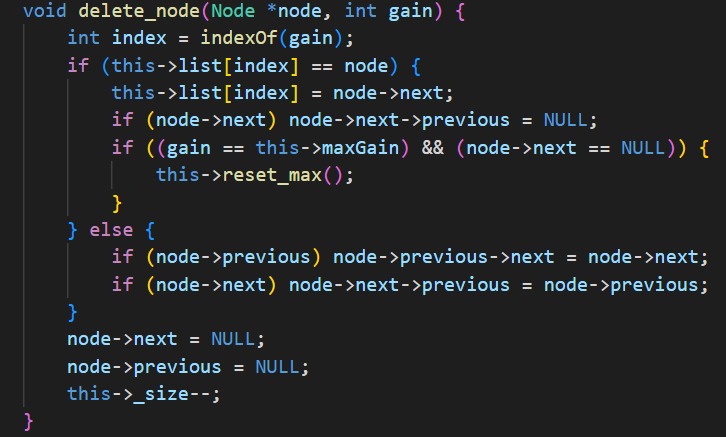
vector<Node \*> list : 真正在負責存東西的vector



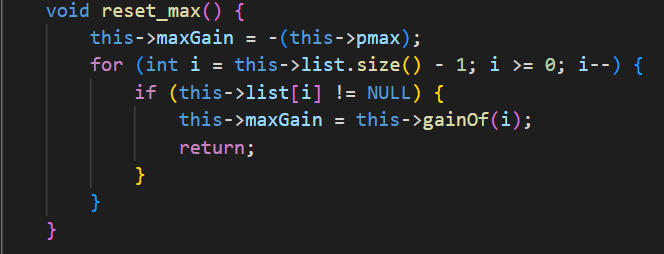
在【新增】或是【刪掉】東西的時候，同時要確保\_size和maxGain都是正確的



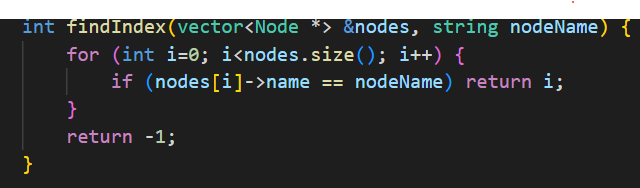
【新增】；node，根據gain值，放到這個list中。



【刪掉】: node，從這個list中gain對應的位置上刪掉。

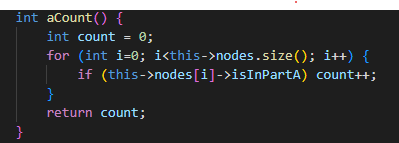


假如我有找到最新最大的maxGain值得話就儲存起來。



如果在nodes裡面，就回答它的位置，不然就回答-1。

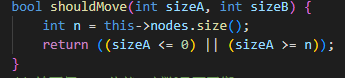
(4) Solver



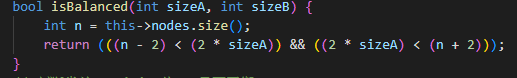
計算當前的partition中有幾個node在part A裡面。



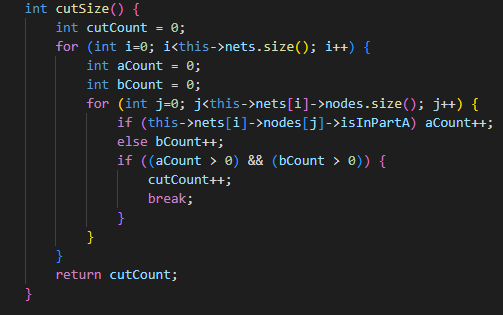
既然我知道partition中有幾個node在part A裡面，我只要扣掉就好。



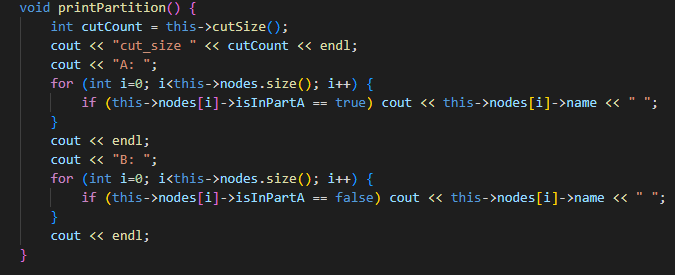
給兩個A, B的值，判斷要不要走。



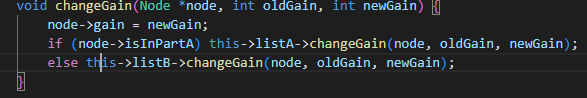
給兩個A, B的值，判斷是否平衡(constrains的判斷式) 【Constraints】。



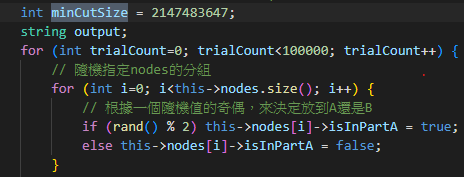
 計算當前的partition有幾個nets被切到。



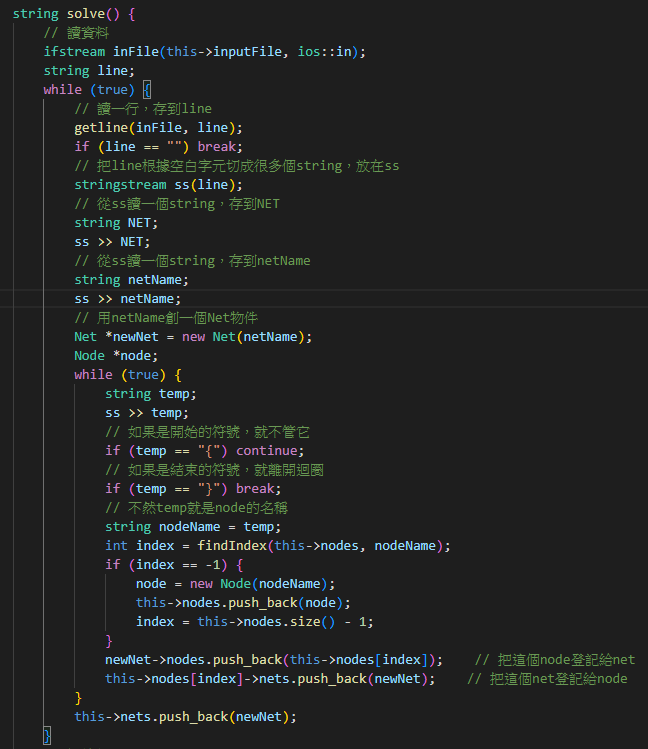
把當前的partition印出來。



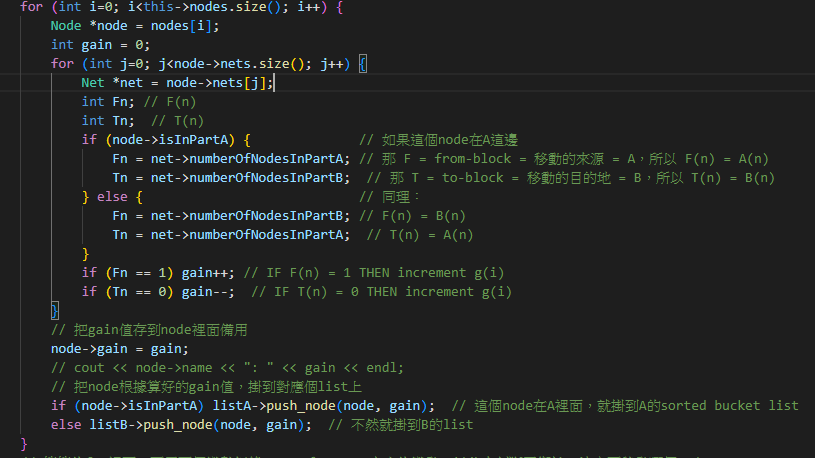
把node根據新的gain值，移到所屬的list的對應的位置上。



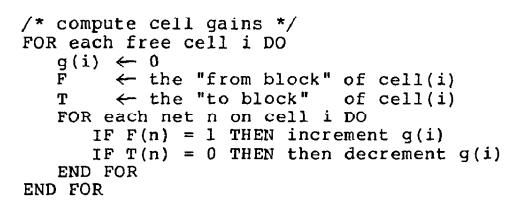
一開始我先把minCutSize設成最大，之後跑n次之後有找比我一開始設的還要小就把它儲存起來。



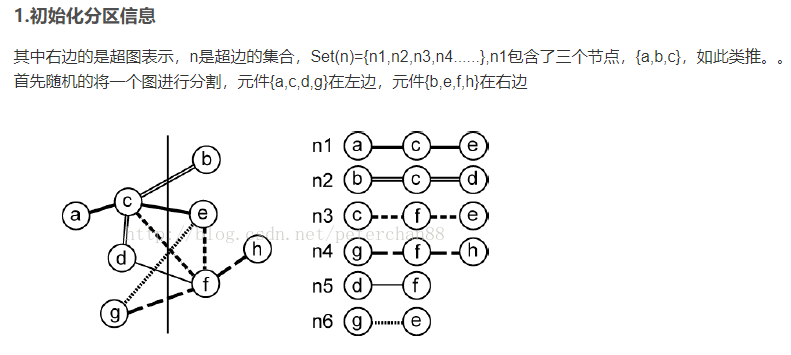
這邊主要是去讀【外面】的資料



我算出gain的值 主要是看下面這邊論文的pseudo code 主要是參考她的資料



Fiduccia-Mattheyses Algorithm



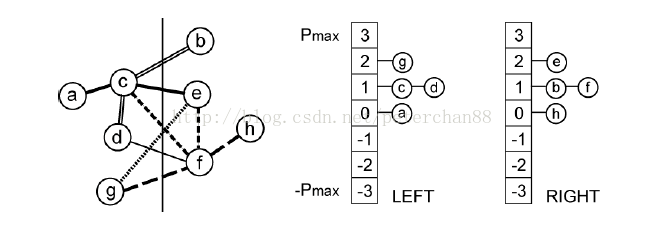
### 2.計算gain值和初始化

對於元件c，c包含在超邊n1={a,c,e}，n3={c,f,e}，n2={b,c,d}中，其中在超邊n3中，只有元件c在左邊的集合中，其他的元件都在右邊的集合中，因此FS(c)+1。並沒有超邊中所有的元組都在左邊，因此TE（c）=0。

FS（i），每當有超邊包含了元件i，而且有且僅有i在元件i所屬的分區（例如元組c在左邊），FS（i）+1，FS（i）初始值為0。TE（i），當有超邊中所有元組都在元件i所屬的分區（例如元組c在左邊），TE（i）+1，TE（0）初始值為0

eg: FS(c) = 1,TE(c) = 0 ==》gain(c) = FS(c)+TE(c) = 1

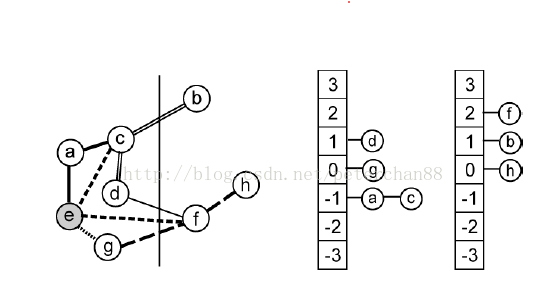
計算出所有元件的gain值：



3.第一次移動

由上面第二步計算出所有元件的gain值，其中元件g和元件e的值都是2是最大的，而且對於g和e都是沒有移動約束的。 ps：移動約束的意思是盡量的使得兩邊的元件數量維持一致，不能有太大的偏差，左邊的數量和右邊的數量要盡量維持相等。這裡根據字母的順序，首先移動元件e。

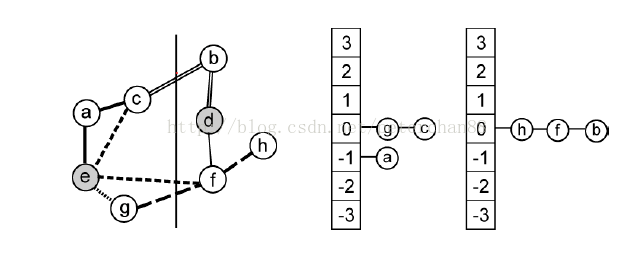
將e從右邊移動到左邊之後，將重新計算與e相關的元件的gain值。與e相關的元件集合{a,c,g,f}，重新計算gain（a） = FS（a）-TE（a） = 0-1 = -1，以此類推gain（c） = -1，gain（g） = 1-1 = 0，gain（f） = 2-0 = 2



4.第二次移動

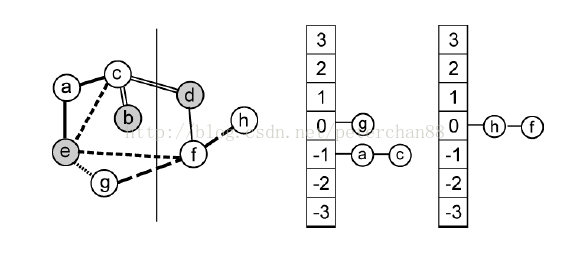
雖然經過了第一次的移動gain（f）的值是最大的，但是f有區域的移動約束（要是移動了f，便導致了左邊的元件數量遠遠大於右邊的元件數量，所以f不能移動），選擇d元件進行移動，隨後計算被影響的元件{b,c,f}，gain（b） = 0-0 = 0，gain（c） = 1-1 = 0，gain（f） = 1-1 = 0

ps：已經移動過的元組就不用重新計算其gain值了，這樣的做法使得算法收斂而且複雜性是線性的！



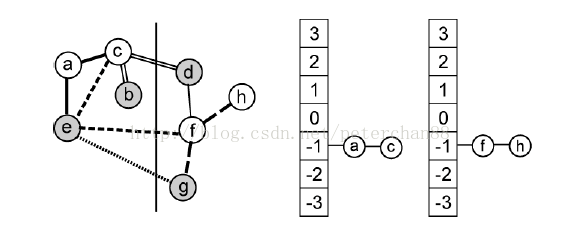
5.第三次移動

經過了兩次的數據移動，g，c，h，f，b的gain值都是0，按照字母的順利，選擇b進行移動，老規矩！計算受影響的元件集合{c}，gain（c） = 0-1 = -1



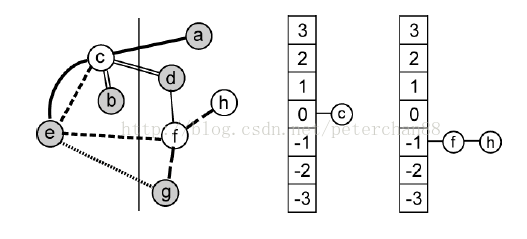
6.第四次移動

g，h，f都是gain值的最大值，根據地域約束條件，選擇g移動，計算受影響的元件{f，h}，gain（f） = 1-2 = -1，gain（h） = 0-1 = -1



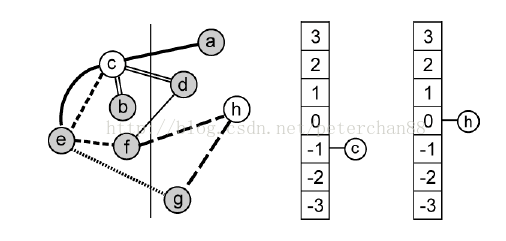
7.第五次移動

按照字母順序，選取a移動，計算受影響的元件集合{c}，gain（c） = 0-0 =0



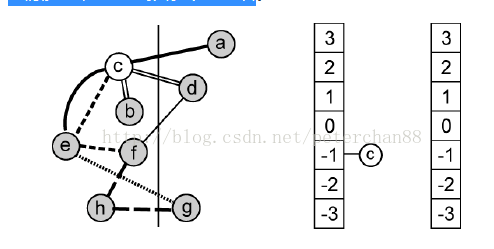
8第六次移動

根據地域約束，選擇f元件進行移動，計算受影響的元件集合{h，c}，gain（h） = 0-0 = 0 gain（c） = 0-1 = -1

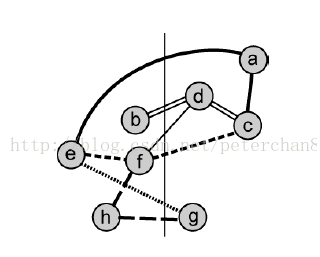


9.第七次移動

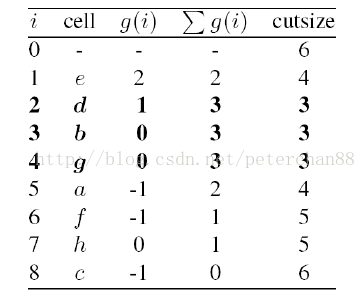
我们移動h，没有受影響的元件集合



10.第八次移動



11.中間數據



i代表的是每一步，cell就是每一步移動的元件cell，g（i）是每一步中的增益gain值，∑g（i）就是將每一步的增益值加起來，cutsize是兩個分區之間的連接數，通俗點來說就是中間斷開了多少條線。 cutsize越少分區的質量越高。在這次的分割中得到了三個最優解分別是第二步，第三步，第四步，cutsize都是3，是最少的！

The hardness of this assignment and how you overcome it。

最難的大概是要update我的gain值跟計算那邊真的卡很久，最後只好去看別人的pseudo code 最後才免強寫出來。雖然case1還是沒辦法到最小的min\_cutsize。

Any suggestions about this programming assignment?

這次作業做完還有 pa4 還要做。做這次作業 好幾天都沒有睡個好覺了，疫情又這麼嚴重，也不敢出門而且又沒有人可以問，加上這幾天又去打疫苗，身體真的不行了。

想到期末有 final project 要做又有 pa4 要做還要準備期末考，三個東西一起來真的會受不了，又沒辦全心全力做一件事情，做事很沒效率，真的會沒時間，這樣沒有一的東西會做得好。懇求老師的法外開恩。

Reference

<https://dl.acm.org/doi/pdf/10.1145/62882.62910?fbclid=IwAR2UMw_YRub5JOGi9HI-jDE2IaDv4Z8zY2GbPG0bexnBdReC1_q9dJD7D1c>