

# Algorithm Programming Assignment 3 Report

## Feedback Arc Set (FAS)

B07901036 陳俊廷

### Concept

大致上而言可以將找 FAS 視為找 MST(maximum spanning tree)，這樣剩下的 edge total weight 便會最少。本次作業大概可分成 undirected graph 與 directed graph 來看，個別的作法如下。

#### Undirected graph

利用 Kruskal's algorithm，較不同的是因為我們需要找到 total weight 最小的 FAS，因此在做 Kruskal 的時候依照 weight 排序後，要改成依 decending order 去做 union。而最後只要把所有不在此 MST(maximum spanning tree)中的 edge 找出即為答案。也就是說，undirected graph 的 fas 是整個 edge set 對 maximum spanning tree 取差集。

#### Directed graph

同樣利用 Kruskals algorithm。但在 directed graph 當中會出現一些問題。例如有三條 edge:  $a \rightarrow b$ ,  $b \rightarrow c$ ,  $a \rightarrow c$ ，此時是沒有 cycle 的，但在 Kruskal 的演算法中  $a, c$  會被視為 representative 相同而視為 cycle，因此  $a \rightarrow c$  便不會在這個 MST 當中。換言之，在 directed graph 當中使用 Kruskal 得到的 MST 並不會是 maximum，還可以增加更多的 edge。因此，我們可以透過在執行完 Kruskal 後，試著把剛剛沒被納入此 MST 的 edge 同樣依照 decending weight order 一條一條加回目前的 MST，並使用 DFS 來測試加回此 edge 後，是否會形成 cycle。並且只須將 weight 為正的加回去即可，因為這樣才能取得 total weight 最小的 FAS。

## Data structure

### Object

Graph: 整個題目定義的  $G$ ，實現全部的 function 並將 edge, node 都存在 graph 中

Node: 一個 vertex，定義一些變數例如 color, fanout 等來方便 graph 計算，也將 Kruskal 中的 p 與 rank 計路成變數。

Edge: 只紀錄 source, target, weight。並且寫 compare function 方便利用 weight sort

### Graph:

Vector<Edge>: 用來紀錄所有 FAS。一開始將全部 edges 讀進，接著在做 Kruskal 的時後，將有被加入 MST 的 edge 從此 vector 中移除，這樣一來對於 undirected 來說可以直接輸出此 edge，對於 directed 來說可以從這個 set 中試著把 edge 加回去並刪除，最後便可得 FAS。因此我從頭到尾都是在對針這個對 weight sorting 過後的 vector 依照 decending order 做操作，會比 linklist, matrix 都還要快。

Node\*: 用動態陣列儲存 vertices。並且用 index 來查找， $O(1)$ 。