

Algorithms

Programming Assignment #3

姓名：莊志淵 學號：r09921006

I. Data structures

主要使用 C++ STL vector 來做為 Graph 與其 transform 的紀錄。

1. G: adjacency list of the given graph.
2. G_T: adjacency list of the inversed graph.

II. Algorithms:

A. Undirected graph:

我選擇使用類似以 Kruskal's algorithm 來對 G 找 minimum spanning tree 的做法，但是把選擇 edge 的優先順序改成由 weight 大的至小的，也就是 maximum spanning tree，該 tree 即為 break cycle 後剩下的 acyclic graph，而沒有選到的 edges 即構成 minimum cost 的 optimal removed edges。

B. Directed graph:

對於 NP-hard 的 minimum feedback arc set problem，我參考 Eades 等人發表的"A fast and effective heuristic for the feedback arc set problem"，裡面介紹了稱為 Algorithm GR 的演算法，是將原先的問題轉換成找 vertex sequence s^* 的問題，讓 G 中的 vertices 由左排到右成一行，而該列中由較右方的 vertex 連到較左方 vertex 的 edges 即為 feedback arc，而 GR algorithm 希望找到一個不錯的順序讓 feedback arc 能夠越少越好。它所採用的方法屬於 greedy algorithm，greedy 由 G 上移除 sinks、sources 以及(outdegree – indegree)最大的 vertices，此移除順序即為所要找的 sequence s，直到 G empty 為止。

```
procedure GR ( $G$  : DiGraph; var  $s$  : VertexSequence);  
 $s_1 \leftarrow \emptyset$ ;  $s_2 \leftarrow \emptyset$ ;  
while  $G \neq \emptyset$  do  
    {while  $G$  contains a sink do  
        {choose a sink  $u$ ;  $s_2 \leftarrow us_2$ ;  $G \leftarrow G - u$ };  
    }  
    {while  $G$  contains a source do  
        {choose a source  $u$ ;  $s_1 \leftarrow s_1u$ ;  $G \leftarrow G - u$ };  
    }  
    choose a vertex  $u$  for which  $\delta(u)$  is a maximum;  
     $s_1 \leftarrow s_1u$ ;  $G \leftarrow G - u$ ;  
 $s \leftarrow s_1s_2$ .
```

論文中有證明 cost 可以小於 $m/2 - n/6$ 。

不過上述的 procedure 只作用在 unweighted 的 edges 上，為了處理 weighted 的部分，我將 degree 的定義改為((input edges 的 weight 總和) – (output edges 的 weight 總和))。