1. **在課程中我們提到了一個被稱為max-flow min-cut theorem的定理，其講述的是network flow的值恰好與min cut的值相符。請證明其正確性。網路上或是教科書上已經有相當多的證明可供參考；但是為了避免單純只是把查到的證明抄下來，請盡量使用中文來描述你的證明，並且過程越細節越好。**

設G = (V,E)是一個網路，而起點s到終點f的最大流為f = s-t，而Gf為f的殘留網路。因為f已經是G的最大流，所以剩餘網路Gf早已不存在一條s到t的路徑，而設集合A為s在Gf上所有可抵達的點，集合B為剩下的其他點，假設|f|=c(A,B)，而|f|又等於A的流出流量總和減A的流入流量總和，因此可得知|f|<=c(A,B)，等號成立於：

(1) 所有A流向B的邊流量都已飽和

(2) 所有B流向A的邊流量不為0

證明(1)：

假設存在不飽和邊(x∈A,y∈B)，即f(x,y)<Cxy，即存在x到y的流量，而(x,y)是Gf的一條邊，這代表s可以只經過A的點通向x再通往y，所以y亦屬於A，產生矛盾。

證明(2)：

假設存在邊(y∈B, x∈A)流量不為0，即f(x,y)>0，剩餘網路裡即存在x到y的流量，而(x,y)是Gf的一條邊，這代表s可以只經過A的點通向x再通往y，所以y亦屬於A，產生矛盾。

故得證|f|=c(A,B)，最大流為最小割。

2. **請分別以pseudocode與連環圖的方式說明何謂network flow problem的Edmonds–Karp algorithm。連環圖的範例就如同課程投影片第286頁或是如https://en.wikipedia.org/wiki/Edmonds%E2%80%93Karp_algorithm 下面所畫的例子那樣。只是本作業限制你的連環圖所用的weighted graph不可與網路上找的到的例子相同，以確保你真的了解Edmonds–Karp algorithm到底怎麼運作。既然有了pseudocode，也請你一併證明Edmonds–Karp algorithm的time complexity是O(|V||E|^2)。**

Pseudocode:

Input: graph G , source s, sink t
Output: maxflow F
EdmondsKarp(G, s, t)
1.set F = 0
2.while(true):
3. min, path = BFS(G, s, t) //從s出發到t結束，min紀錄路線每條邊中的最小flow，path紀錄所走路線
4. if(min = 0) break //終止迴圈
5. F = F+min
6. for each (u, v) in path:
7. G[u, v] = G[u, v] – min //路線正向邊扣除流量min
8. G[v, u] = G[v, u] + min //路線反向邊扣除流量min
9.return F

Time compactity:

V 為點總數 E 為邊總數

首先我們得知 DFS 尋找路線的時間複雜度為 O(|E|)

而每經過一條邊都會長出一條反向邊，但也不會使總邊數超過原本 2 倍，O(2|E|) = O(|E|)

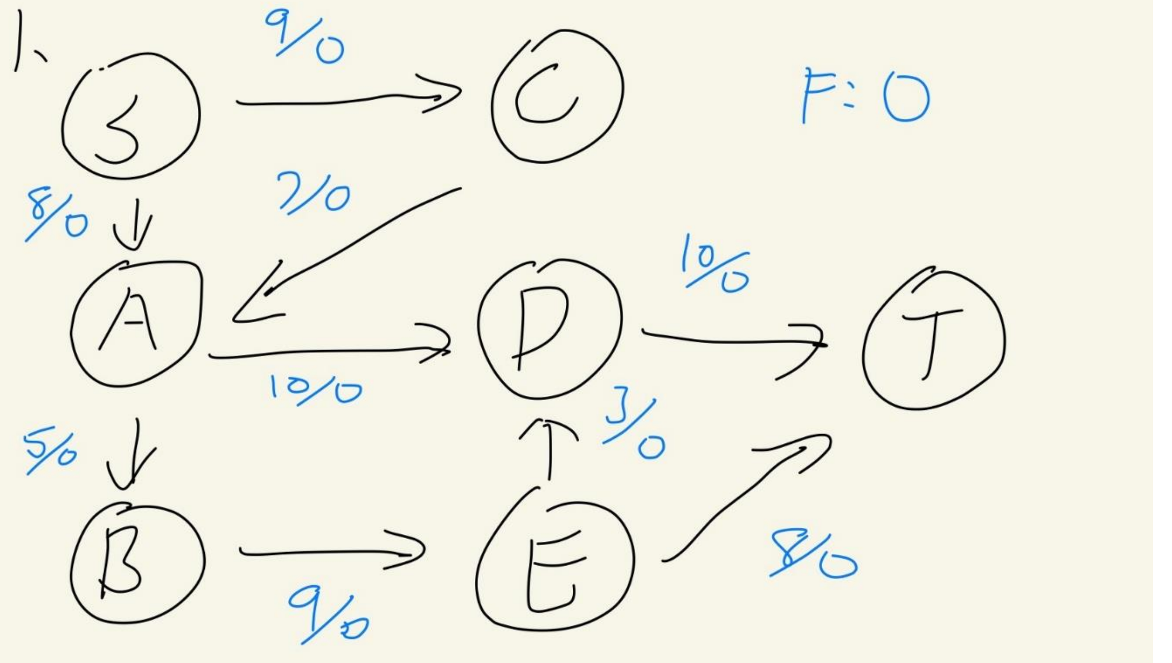
而要證明更新路線的正/反向邊流量時時間複雜度為O(|V||E|)，首先需要證明更新路線不會導致s到任何一點v的最短路線距離減少，在殘留圖Gf經過一次BFS增廣變成Gf' 後，不會導致s到任何一點v的最短路線長度遞減，即d'(s, v) >= d(s, v)：

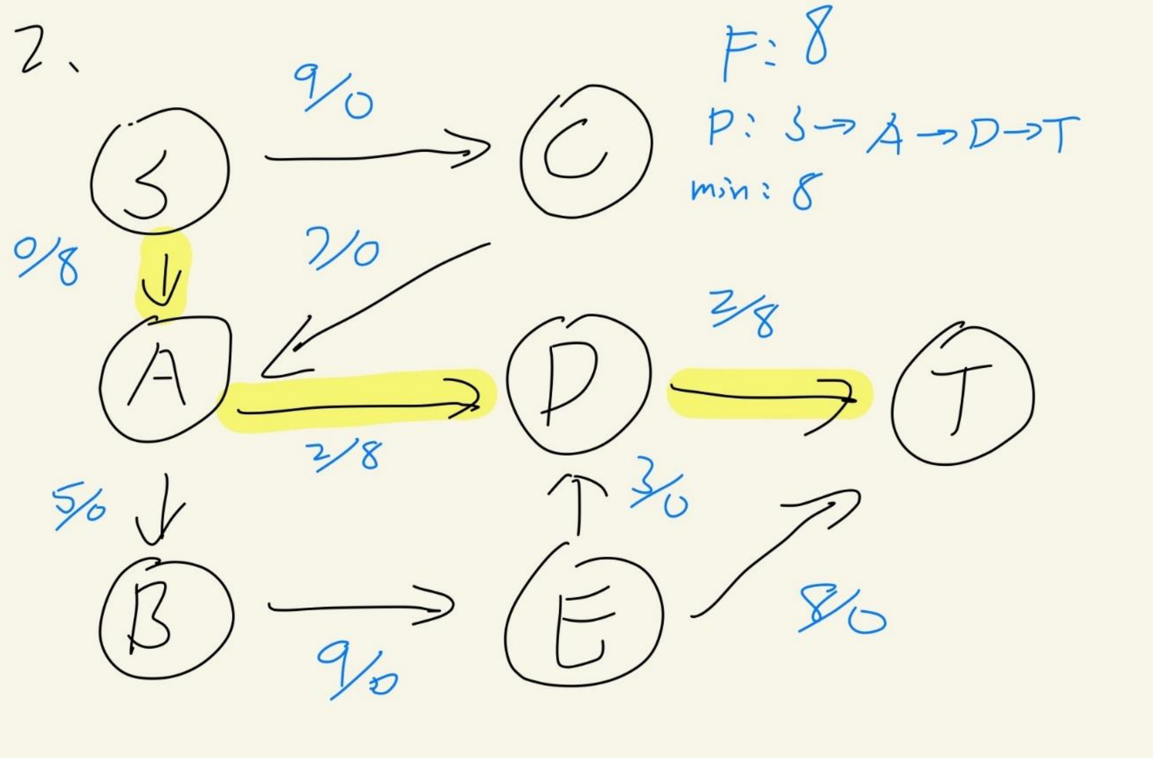
(1)設經過增廣後，v為到s的最短路徑變小中的點之最近者 即d'(s, v) < d(s, v)
(2)設u為Gf' 中v的前一個頂點 即d'(s, u) + 1 = d'(s, v)
(3)u比v更靠近s，但未成為最短路徑變小中的點之最近者 即d'(s, u) >= d(s, u)
(4)設(u, v)屬於Ef(未使用邊)，已知s到v的最短路徑為d(s, v) 即d(s, u) + 1 = d(s, v)
結合(1)(2)(3)即可得到 d(s, u)+1 < d(s, v) 但與(4)產生矛盾
因此可知(u, v)不屬於Ef，但按照(2)所述(u, v)屬於Ef，所以在Gf的增廣上必定經過(v, u)且飽和，導致Gf' 中產生了反向邊(u, v)，可知存在著d(s, u) = d(s, v) + 1
此項又與(4)矛盾，代表(1)的假設錯誤，用反證法可得知d'(s, v) >= d(s, v)

而根據上述證明
又可得到 d'(s, u) = d'(s, v) + 1 >= d(s, v) + 1 = d(s, u) + 2
s離最遠點的距離不超過|V|-1，所以(u, v)可以成為飽和邊次數最多為(|V|-1)/2，每次增廣至少有一條邊飽和，根據前述總邊數不超過原本2倍，所以增廣次數最多為2|E| * (|V|-1)/2，更新路線的時間複雜度即為O(|V||E|)

所以Edmonds-Karp Algorithm的時間複雜度為O(|E|) * O(|V||E|) = O(|V||E|^2)

連環圖：


1、
F: 0
S
C
A
D
T
B
E
9/0
7/0
8/0
10/0
10/0
3/0
5/0
8/0
9/0

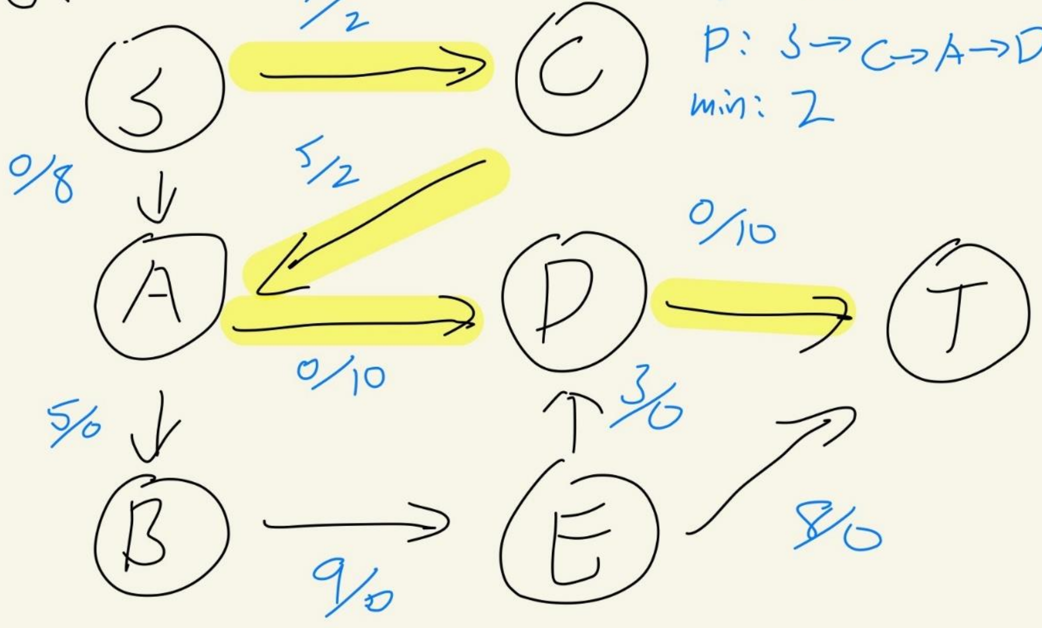

2、
F: 8
P: S→A→D→T
min: 8
S
C
A
D
T
B
E
9/0
7/0
0/8
2/8
2/8
3/0
5/0
8/0
9/0

3.

F: 10

p: S→C→A→D→T

min: 2

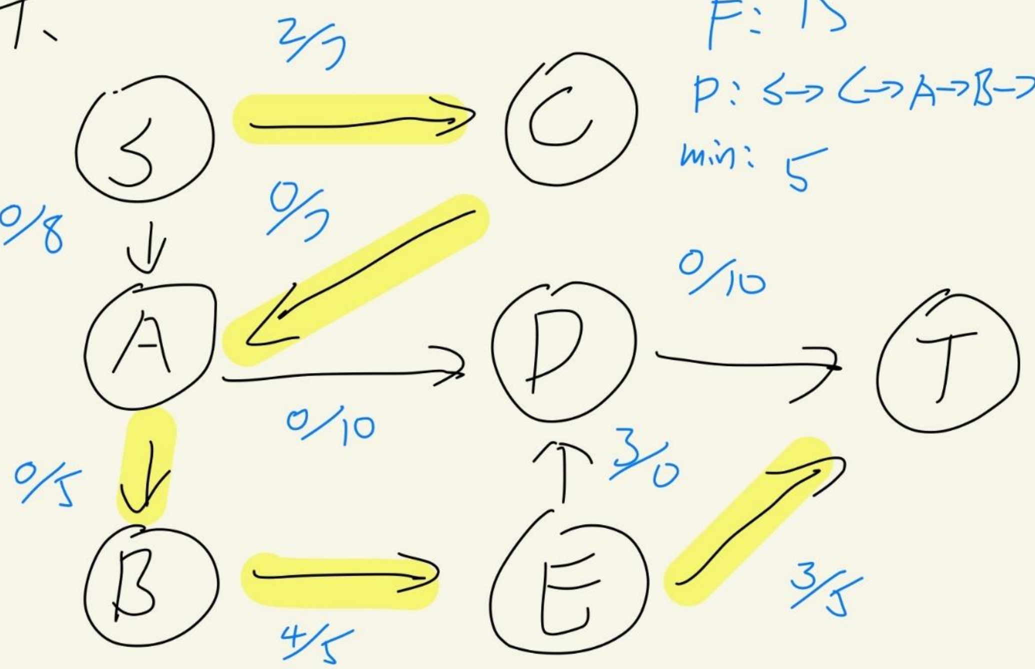


4.

F: 15

p: S→C→A→B→E→T

min: 5



5、

F: 15

P: 已無 augmenting path

min: 0