

# Data Structure Hw4

# 1. Graph 的基本操作

- void addVertex(int a)
- void deleteVertex(int a)
- void addEdge(int a, int b, int w)
- void deleteEdge(int a, int b)
- void connectedComponents(void)
- void Dijkstra (int a, int b)
- void BellmanFord (int a, int b)

# 格式規範

- `void addVertex(int a)`:將編號a的點加入G中

參數a	結果
$a < 0$ or $a > 100$	印出"add an invalid vertex"
0~100	(1)若a點在G中，印出"vertex <b>a</b> is already in G." (2)若a點不在G中，將a點加入G

- `void deleteVertex(int a)` :將編號a的點從G中移除

參數a	結果
$a < 0$ or $a > 100$	印出"delete an invalid vertex"
0~100	(1)若a點不在G中，印出"vertex <b>a</b> isn't in G." (2)若a點在G中，將a點和所有與a點相鄰的邊移除

# 格式規範

- `void addEdge (int a, int b, int w)`:將權重 $w$ 且 $a$ 指向 $b$ 的邊加入 $G$ 中

參數a	參數b	參數w	結果
a不在G中	不限	不限	a 不在G中，印出"add an invalid edge"
不限	b不在G中	不限	b不在G中，印出"add an invalid edge"
a在G中	b在G中	不限	(1)<a,b>已在G中，w取代原<a,b>上的權重 (2)將權重w的邊<a,b>加入G中

- `void deleteEdge(int a, int b)`:移除 $G$ 中 $a$ 指向 $b$ 的邊

參數a	參數b	結果
a不在G中	不限	a 不在G中，印出"delete an invalid edge"
不限	b不在G中	b不在G中，印出" delete an invalid edge"
a在G中	b在G中	(1)<a,b>在G中，將邊<a,b>從G中移除 (2)<a,b>不在G中，印"fail to delete edge <a,b>"

# 格式規範

- void connectedComponents (void):計算G中strongly connected components的個數，忽略權重

參數	結果
無	印出G中strongly connected components的個數，ex:"connected components: 3"

Hint: Strongly connected components

1. 對 $G=(V,E)$ 做DFS(起始點不重要)，求各vertex finished time的順序(小→大)
2. 求 $G^T = (V, E^T)$ ,  $E^T = \{(u, v) : (v, u) \in E\}$
3. 對 $G^T$ 做DFS，起始點依步驟一finished time的順序反向(大→小)

Hint:用stack實做DFS，用timestamp紀錄時間，則當push a vertex  $u$  to the stack時的time stamp稱為 $u$ 的discovered time，而當pop a vertex  $u$  from the stack時的time stamp稱為 $u$ 的finished time

# 格式規範

- void Dijkstra (int a, int b)

參數a	參數b	結果
a在G中	b在G中	使用Dijkstra's algorithm求出點a到點b的最短路徑， (1)G中含有負邊(不用求有多少條負邊)，印出" G contains a negative edge." (2)a無法到達b，印出" Dijkstra: no solution for a->b" (3)a可到達b，印出最短路徑和總權重，ex:"a->c->b total: 10"
a不在G中	不限	印出" an invalid vertex"
不限	b不在G中	印出" an invalid vertex"

# 格式規範

- void BellmanFord (int a, int b)

參數a	參數b	結果
a在G中	b在G中	使用Bellman–Ford algorithm求出點a到點b的最短路徑， (1) G中含有負cycle (不用求有多少個負cycle)，印出” G contains a cycle of negative length.” (2)a無法到達b，印出” BellmanFord: no solution for a->b” (3)a可到達b，印出最短路徑和總權重，ex:”a->c->b total: 10”
a不在G中	不限	印出” an invalid vertex”
不限	b不在G中	印出” an invalid vertex”

Hint:

BellmanFord執行第 $|V|$ 圈後，若a點到各點的距離有減少，即代表G中含負cycle

# 執行結果1

印出的錯誤資訊時紅字字元要換成對應的數字  
同個指令的輸出要在同一行

```
input.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
addVertex 0
addVertex 1
addVertex 2
addVertex 0
addEdge 0 1 5
addEdge 1 2 -4
addEdge 0 2 4
Dijkstra 0 2
addEdge 0 1 2
addEdge 1 2 1
Dijkstra 0 2
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19044.1645]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
D:\>hw4_1.exe < input.txt
vertex 0 is already in G.
G contains a negative edge.
0->1->2 total:3
D:\>
```



# 執行結果2

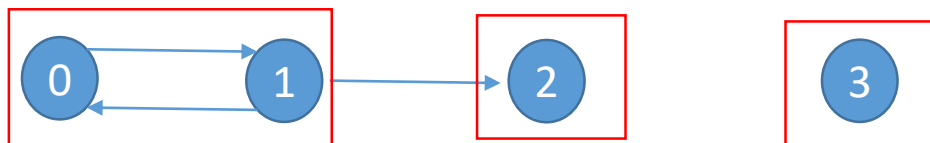
```
input2.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
connectedComponents
addVertex 0
addEdge 0 1 1
addVertex 1
addVertex 2
addEdge 0 1 1
addEdge 1 2 1
addEdge 1 0 -2
BellmanFord 0 2
addEdge 1 0 2
BellmanFord 0 2
addVertex 3
connectedComponents
```

C:\Windows\System32\cmd.exe

```
Microsoft Windows [版本 10.0.19044.1706]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

D:\>hw4_1.exe < input2.txt
connected components: 0
add an invalid edge
G contains a cycle of negative length.
0->1->2 total:2
connectedComponents: 3

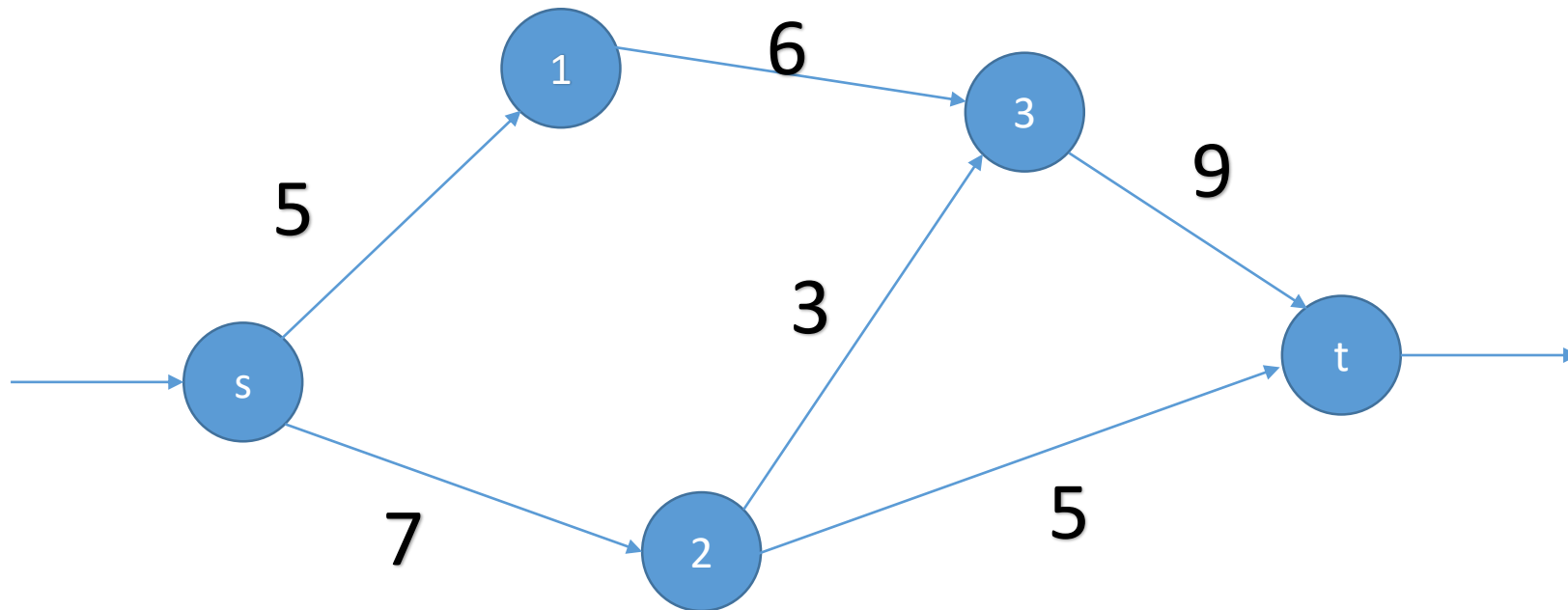
D:\>
```



3 個 connected components

## 2. Max-flow min-cut

- Flow network(流量網路)  $G=(V,E)$  是一個有向圖，每一邊  $(u,v) \in E$  均有 Capacity(容量)  $c(u,v) > 0$ 。如  $c(u,v)=0$  即代表  $(u,v) \notin E$ 。



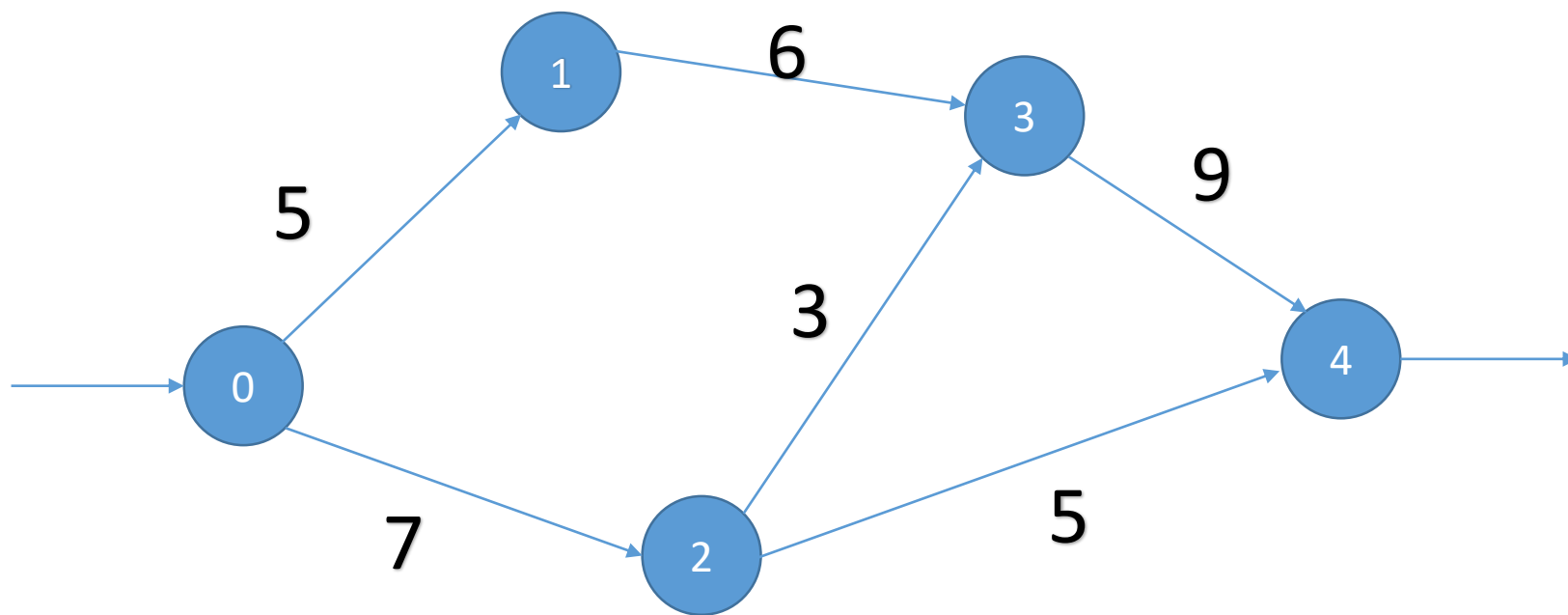
- 令 $s$ 為Source vertex， $t$ 為Sink vertex。一個Flow(流量)係一函數 $f: V \times V \rightarrow \mathbb{R}$ ，對任兩點 $u, v$ 而言滿足下列性質：
  - Capacity constraint:  $f(u, v) \leq c(u, v)$
  - Skew symmetric:  $f(u, v) = -f(v, u)$
  - Flow conservation: 若 $u \in V - \{s, t\}$ ，則 $\sum_{w \in V} f(u, w) = 0$ 。
- $|f| = \sum_{v \in V} f(s, v)$ 稱作流量 $f$ 的值。

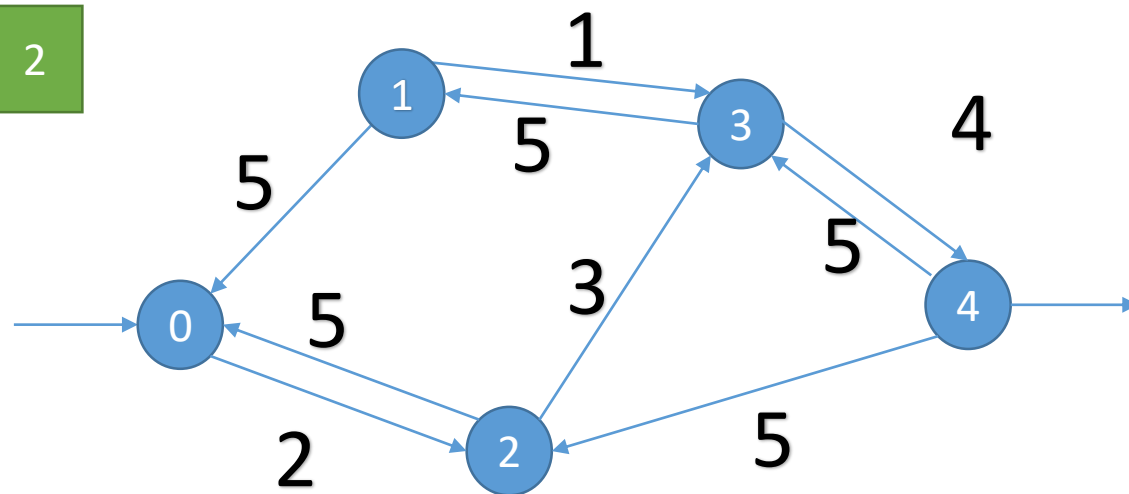
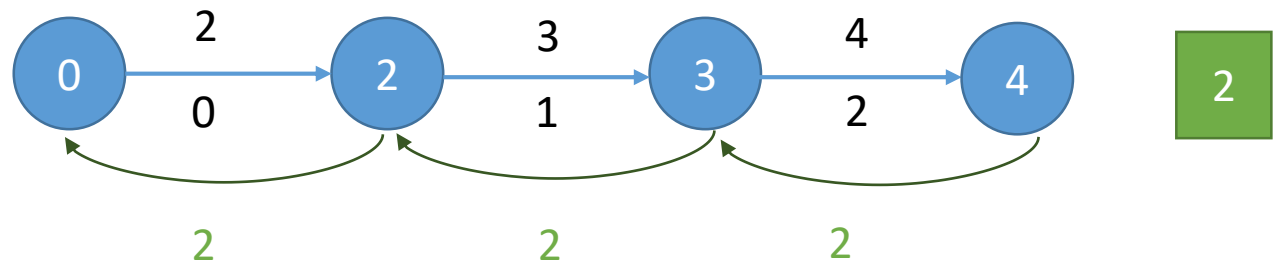
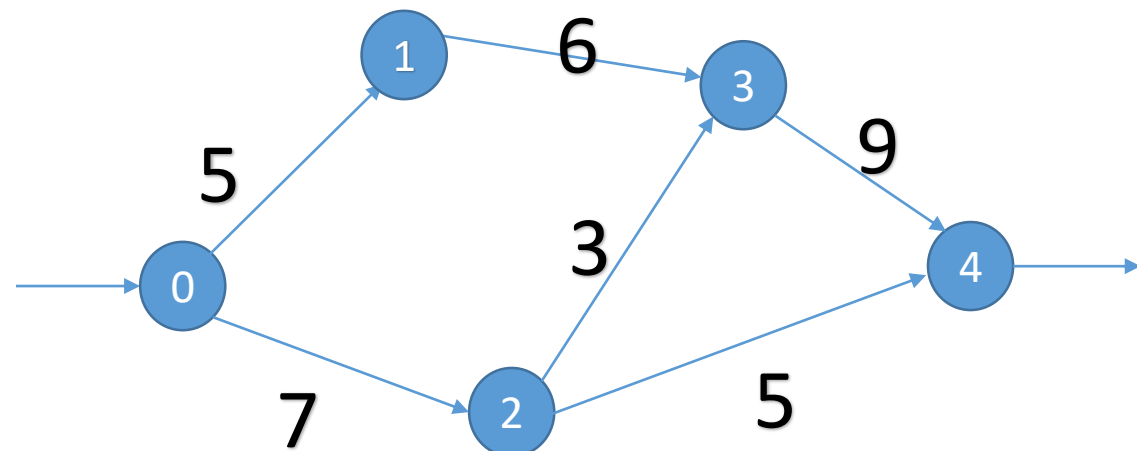
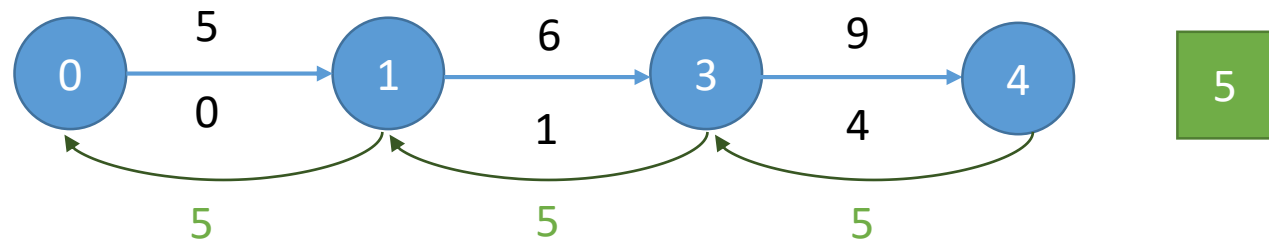
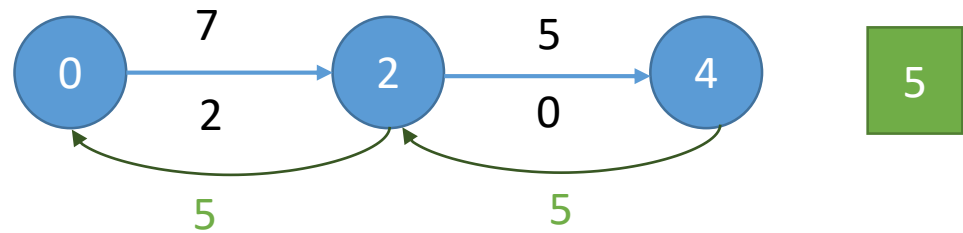
# Residual Network與Augmenting Path

- 由一Flow network  $G$ 及一Flow  $f$ 所導出的Residual network  $G_f$ 為一個Flow network，其Capacity  $c_f(u,v)=c(u,v)-f(u,v)$ 。
- 一個Flow network  $G$ 及Flow  $f$ 所導出的Augmenting path即是Residual network  $G_f$ 上一個 $s \rightarrow t$ 的路徑 $p$ 。
- 如果一Flow network找到的Augmenting path代表可以找到一個Flow  $f$ ，其值大於0。
- 如一Flow network  $G$ 及一Flow  $f$ 所導出的Residual network  $G_f$ ，以找不到任何Augmenting path，則 $f$ 是最大流量。

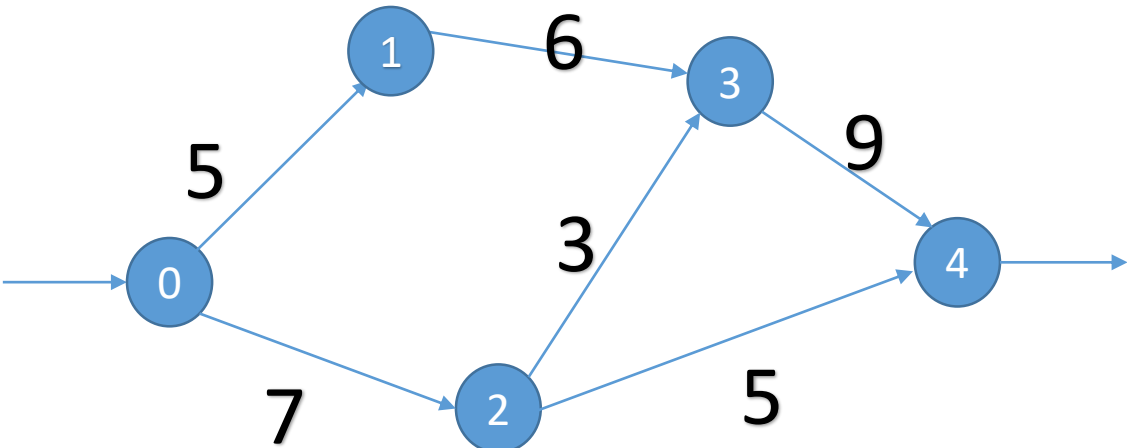
Hint: 可以用BFS或DFS找Augmenting Path

- 給定水流管線圖G，水會從起點(0)流向終點(4)，求出最大流量f

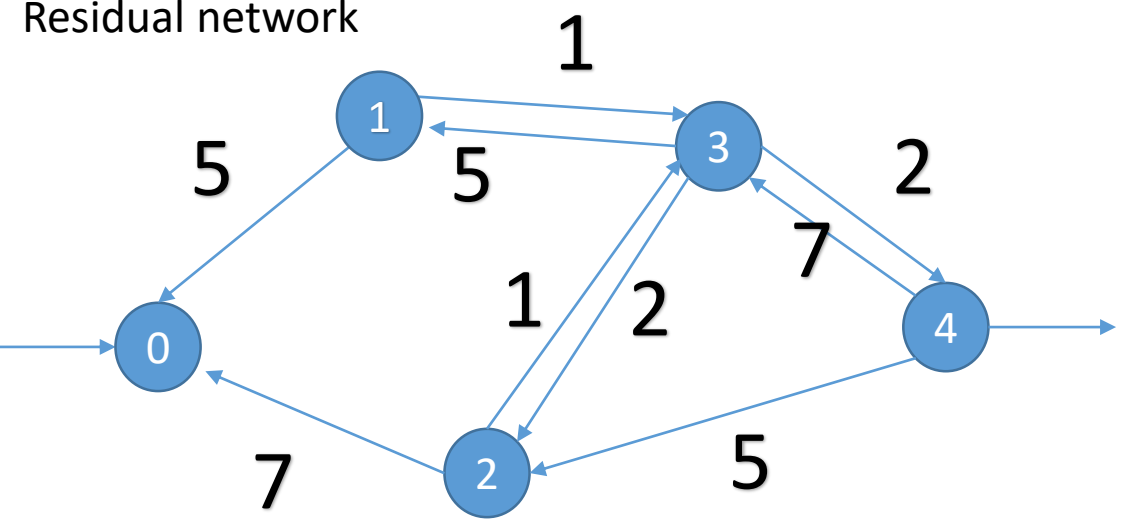




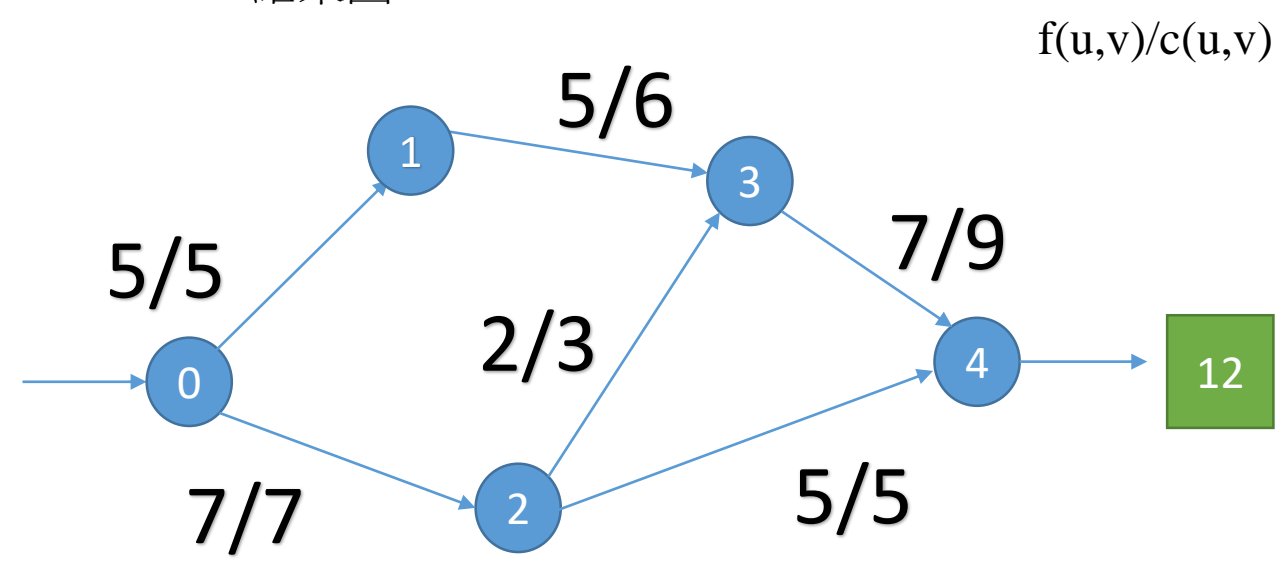
原圖



Residual network



結果圖:Flow network



# 格式規範

- 輸入:

Ex:

5

0 1 5

0 2 7

1 3 6

2 3 3

2 4 5

3 4 9

n:vertex個數  
(起點:0, 終點:n-1)

a b c  
(<a,b>的容量為c)

- 輸出:

邊的表示式、流量和容量(請以空格或  
tab隔開)

Ex:

<0,1> 5 5

<0,2> 7 7

<1,3> 5 6

<2,3> 2 3

<2,4> 5 5

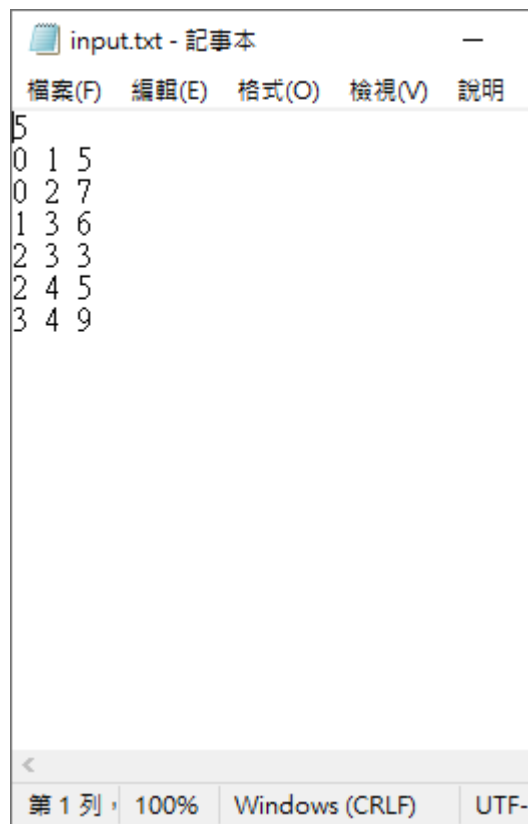
<3,4> 7 9

total:12

流量和容量都為非負整數

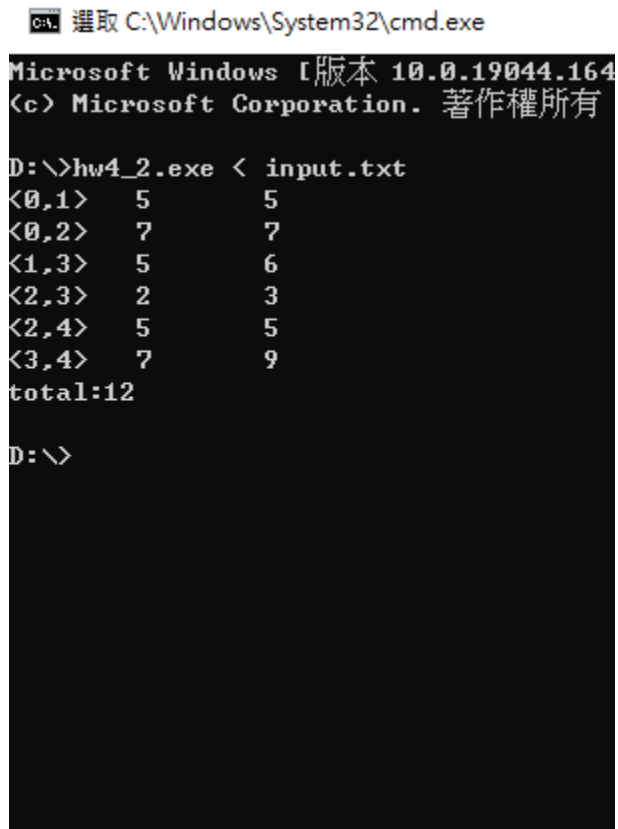


# 執行結果1



```
input.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
5
0 1 5
0 2 7
1 3 6
2 3 3
2 4 5
3 4 9
```

第 1 列, 100% Windows (CRLF) UTF-



```
C:\> 選取 C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19044.164]
(c) Microsoft Corporation. 著作權所有

D:\> hw4_2.exe < input.txt
<0,1> 5 5
<0,2> 7 7
<1,3> 5 6
<2,3> 2 3
<2,4> 5 5
<3,4> 7 9
total:12

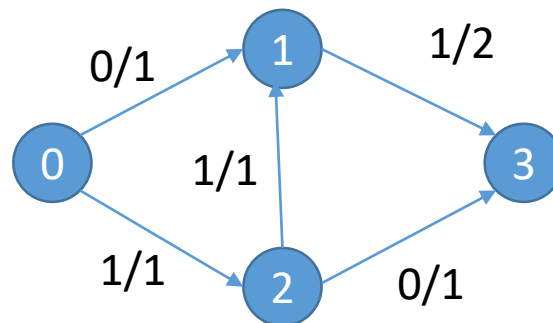
D:\>
```

# 執行結果2

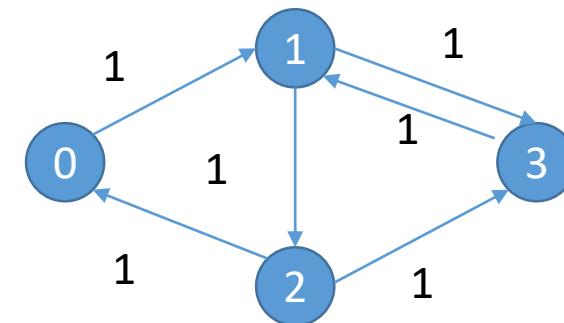
```
input2.txt - 記事本
檔案(F) 編輯(E) 格式(O)
4|
0 1 1
0 2 1
1 3 2
2 1 1
2 3 1

D:\>hw4_2.exe < input2.txt
<0,1> 1 1
<0,2> 1 1
<1,3> 1 2
<2,1> 0 1
<2,3> 1 1
total:2
D:\>
```

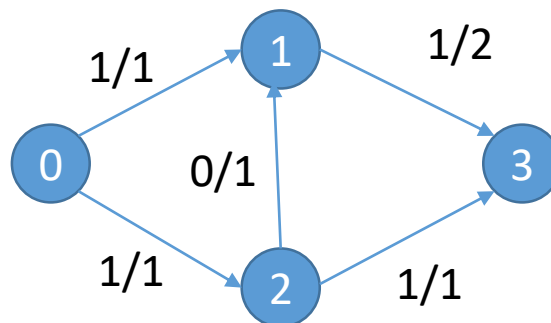
更新path:0→2→1→3後的  
Flow network



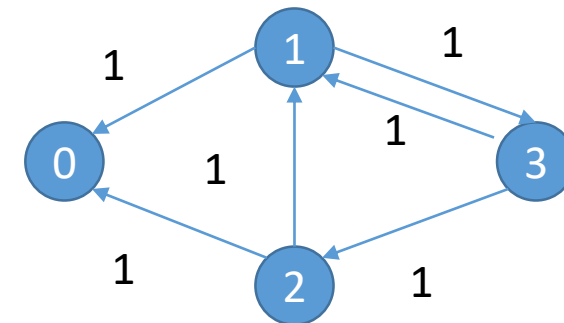
更新path:0→2→1→3後的  
residual network



更新path:0→1→2→3後的  
Flow network



更新0→1→2→3後的  
residual network



當 $f(u,v)>0$ 時才有可能產生逆向水流與原方向的水流抵銷

# 注意事項:

1. 請使用C和C++程式語言完成作業
2. 有任何問題請寄信給助教
3. 禁止使用C++ STL函式庫(包括vector)，否則0分，可以用<Algorithm>但這次作業不能直接呼叫排序相關函式
4. 函式的參數和回傳值不限，只要能夠正確讀檔和執行功能即可
5. 若因語法或版本問題導致助教無法執行程式，會請同學帶著電腦來實驗室demo，不會直接拿不到分數
6. 第一題不含loop
7. 若<a,b>和<a,c>的權重相同，編號b小於編號c，則在權重相同的情況下優先選擇相鄰點編號較小的邊<a,b>
8. 抓到抄襲，直接0分
9. 若code編譯出的執行檔與繳交的執行檔不符，會以前者為評分標準

# 評分標準:

- Graph 的基本操作(50%)
- Max-flow min-cut(50%)

# 作業繳交

E12345678\_王小明(.rar/.zip)

hw4\_1

hw4\_1(.c/.cpp)  
\*(.h/others)

hw4\_2

hw4\_2(.c/.cpp)  
\*(.h/others)

exe\_file

hw4\_1(.exe/.out/.exec)  
hw4\_2(.exe/.out/.exec)

格式錯誤該次作業總成績扣十分

繳交期限:

6/6(一) 23:59:00前至 moodle 上傳作業

不接受任何理由之遲交