最大流問題

Maximum Flow

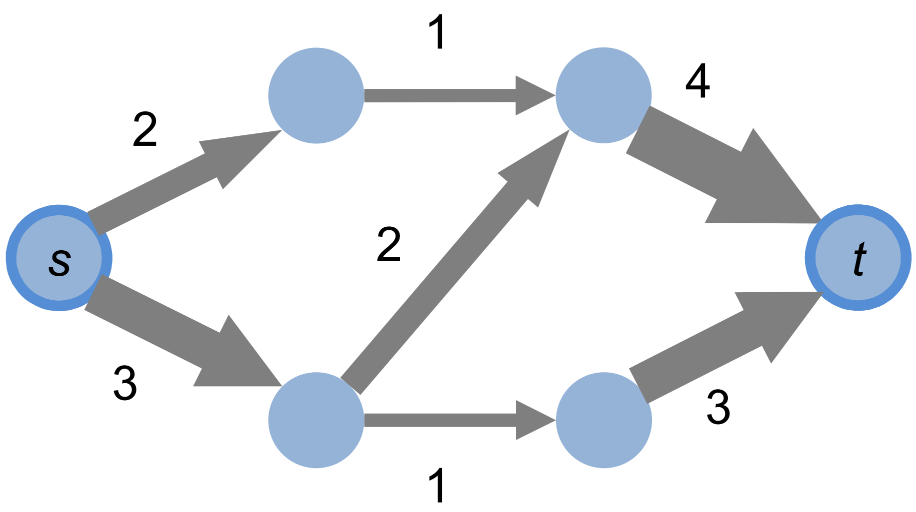
Chapter 26

Mei-Chen Yeh

5/18 (四)

Slides are modified from Prof. Lu START

Network



- A directed graph G, each of the edges has a *capacity*, $c(u, v) \geq 0$.

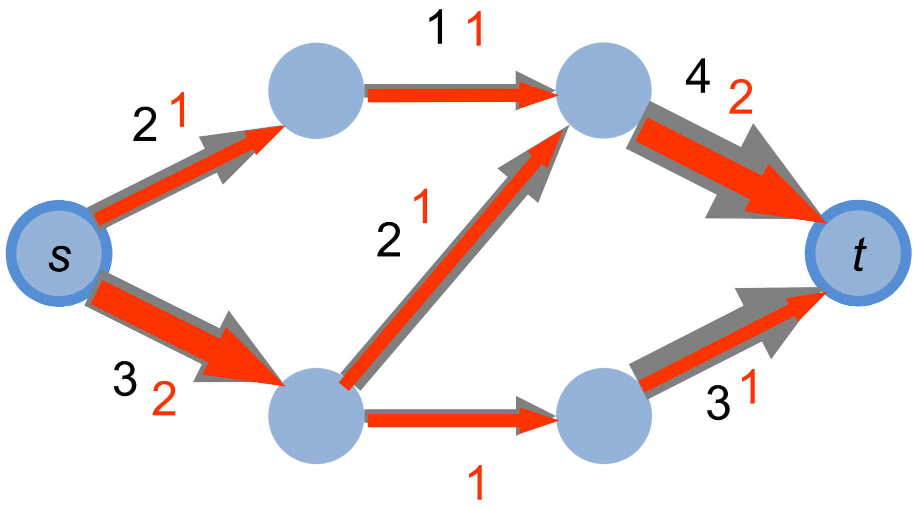
這裡的權重值是有意義的，是capacity

- Two distinguished nodes in G: a *source* (s) and a *sink* (t)

在這個圖有兩個特殊的節點：source, sink

Flow

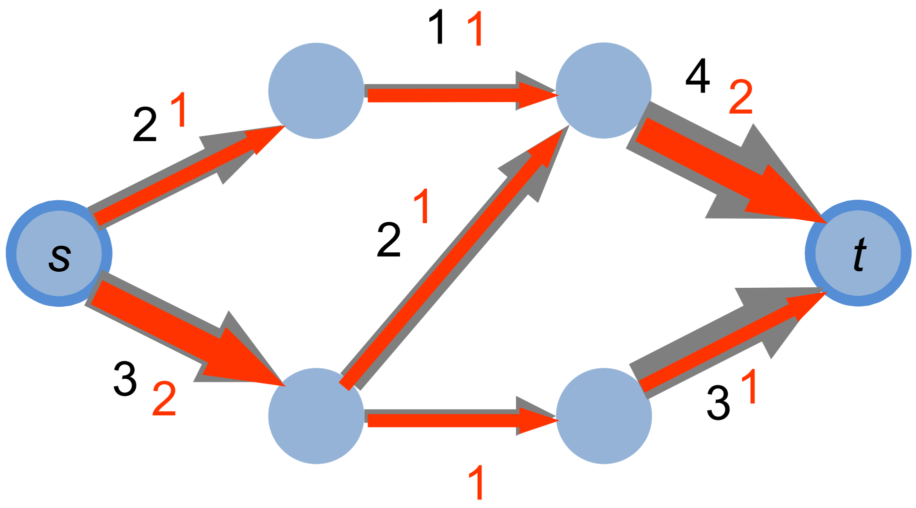
從起點流出去的是1+2 = 3



這個 3 就會流到終點

- A flow *f*(*s*, *t*) of the network (*G*, *s*, *t*) is a weighted subgraph of *G* that satisfies the
  流量不能超過capacity　　要守恆，流多少進來就流多少出去
  capacity constraint and *flow conservation*.
  flow是subgraph，要滿足這兩個條件
- The value of a flow is the sum of weights of the outgoing edges of *s* in *f*.

Capacity constraint

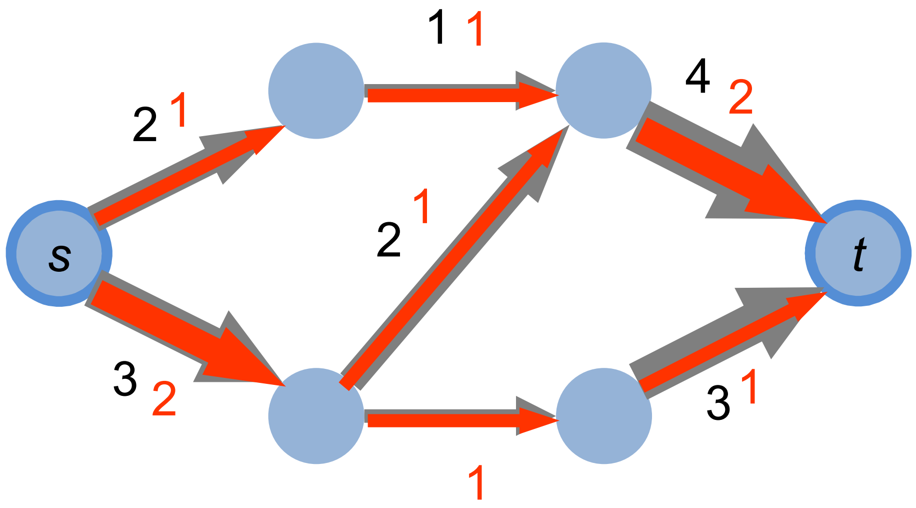


For all $u, v \in V$, we require

$$0 \le f(u,v) \le c(u,v).$$

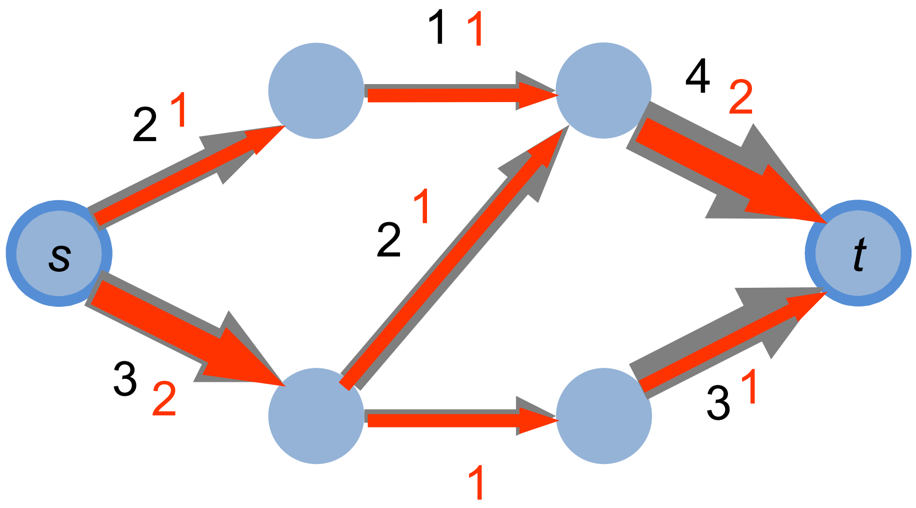
流量不能超過capacity

Capacity constraint



- The amount of flow passing through an edge has to be **no more than** the capacity of the edge.

Flow conservation

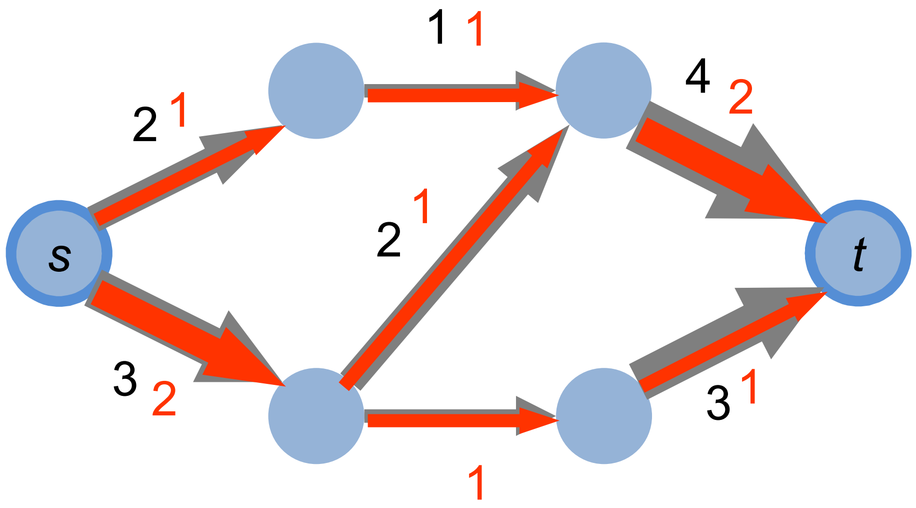


For all $v \in V - \{s, t\}$, we require

$$\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v).$$

起點流多少出去，終點就會流多少回來

Flow conservation

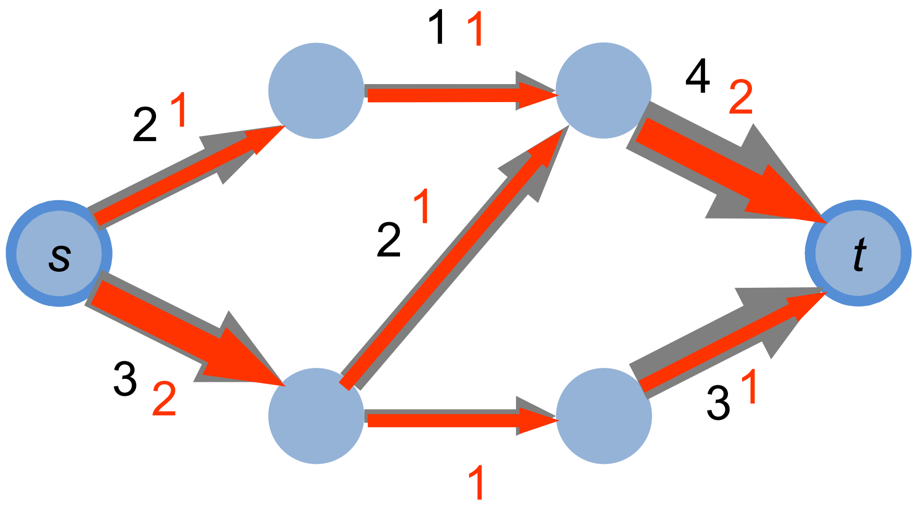


- For any node other than *s* and *t*, the amount of flow entering *u* has to be equal to the amount of flow leaving *u*.

"flow in equals flow out"

在現實的運用是在貨運上，或是匹配問題

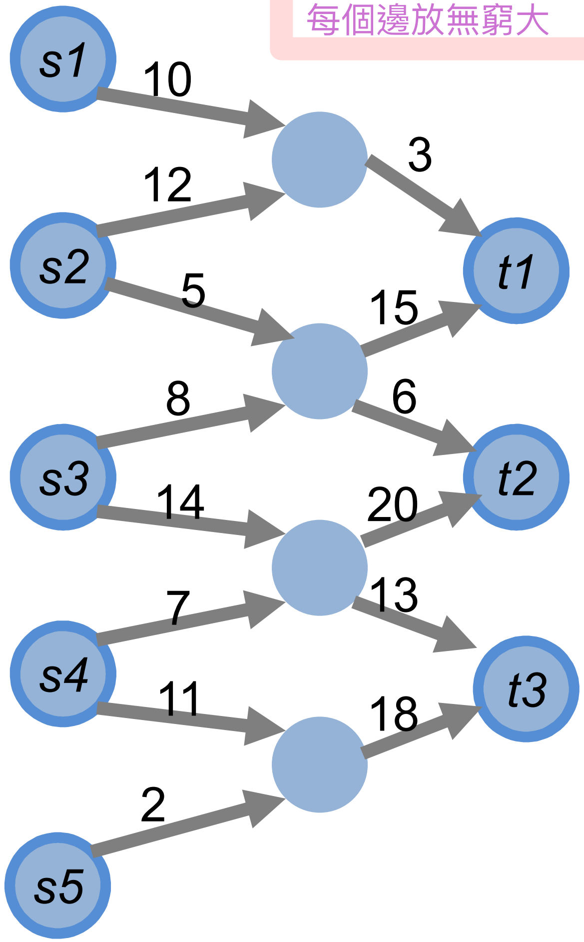
The maximum-flow problem

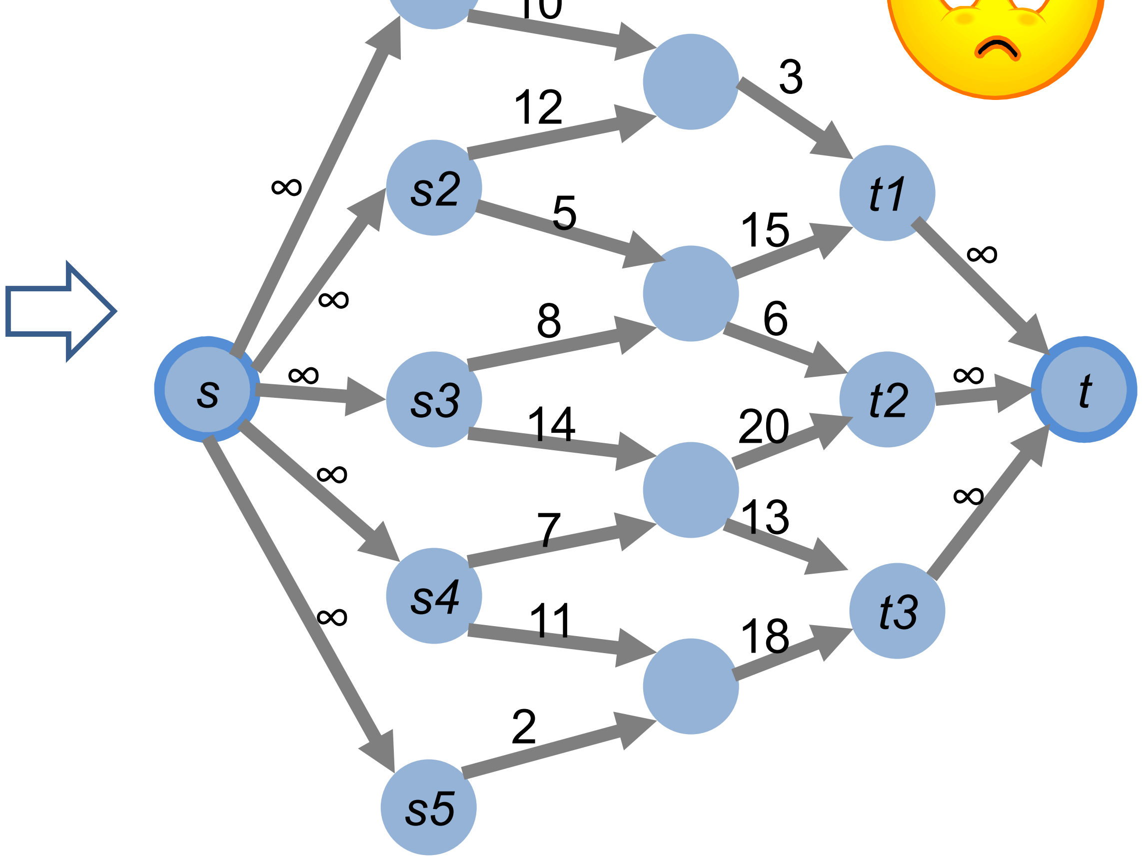


- Given a network *G* with source *s* and sink *t*, find a flow of maximum value.

A network with multiple sources and sinks?

多起點多終點怎麼辦？
加一個super source和super sink，
在前後各多加一個節點當起點和終點，
每個邊放無窮大





最大流問題最基本的演算法

The Ford-Fulkerson method

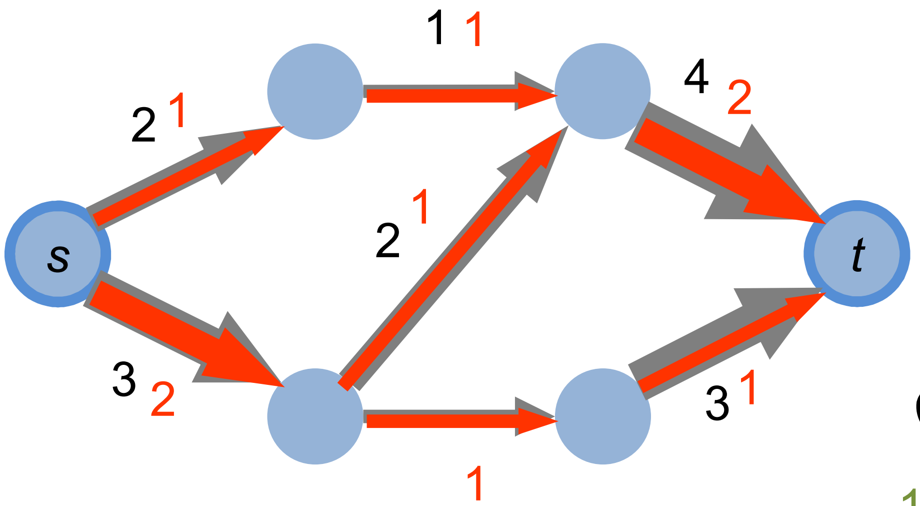
FORD-FULKERSON-METHOD(G, s, t)

1. initialize flow f to 0 先把flow初始化成0
2. **while** there exists an augmenting path p in the residual network G_f, augment flow f along p

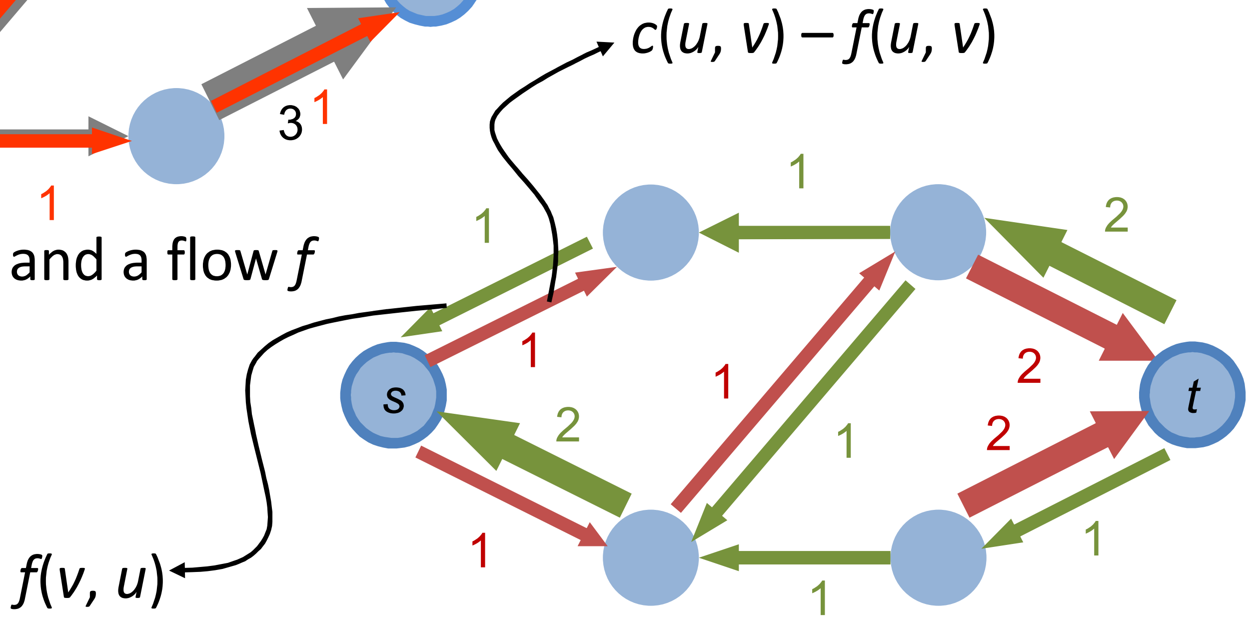
用 while loop，
只要在residual network上面發現還有augmenting path，
就把它加上去，直到沒有augmenting path為止

3. **return** f 累積起來的值就會是最大流

Residual network



a network G and a flow f

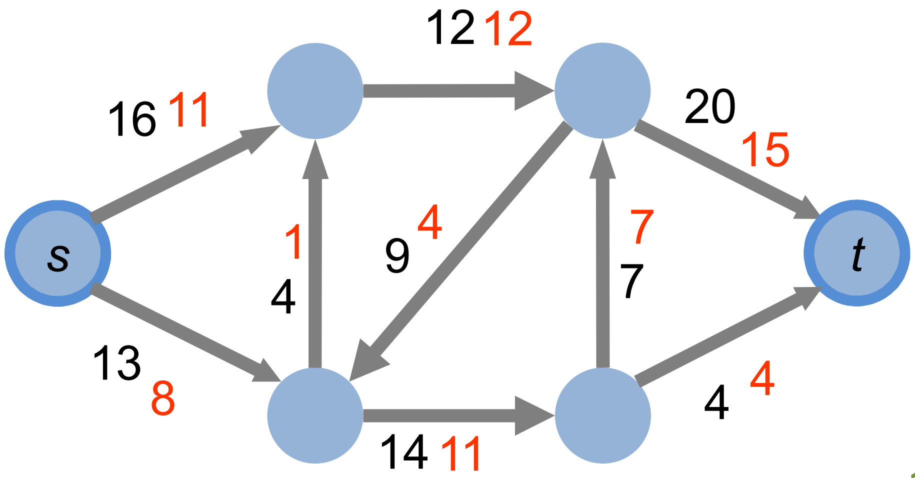


The residual network G_f of G and f

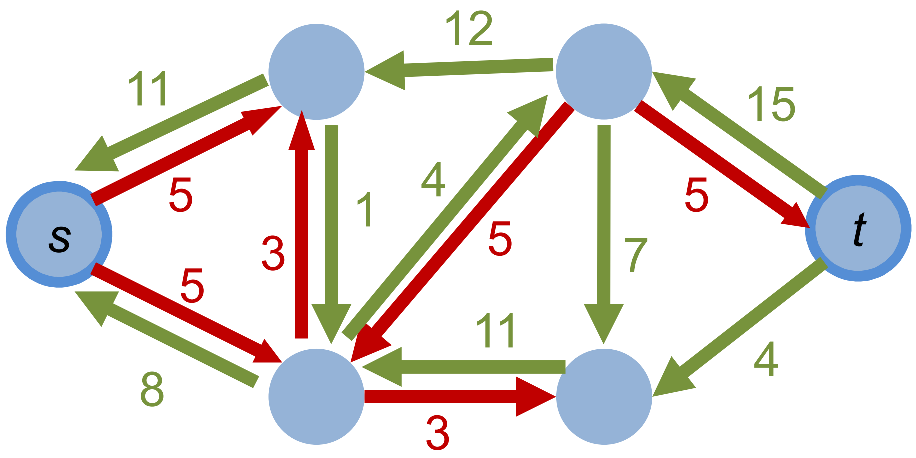
紅色代表 capacity 還剩多少
綠色代表 原本的flow的反向

Residual network

- One more example:



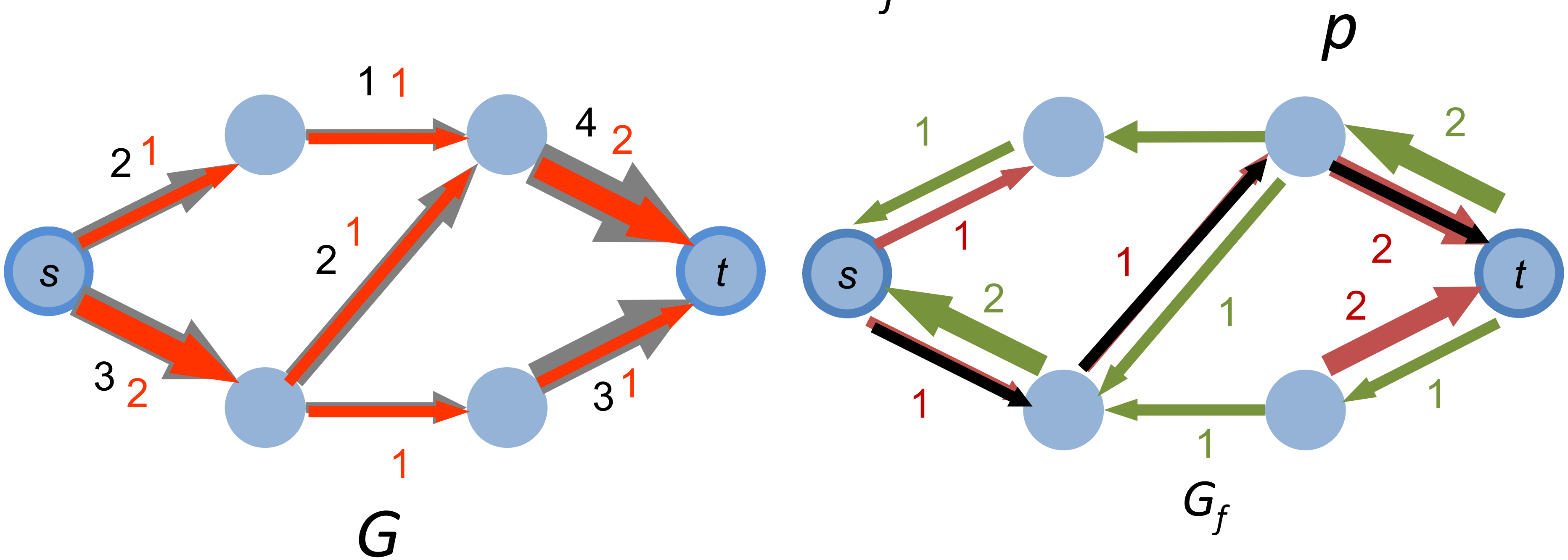
紅色代表 capacity 還剩多少
綠色代表 原本的flow的反向



Augmenting path

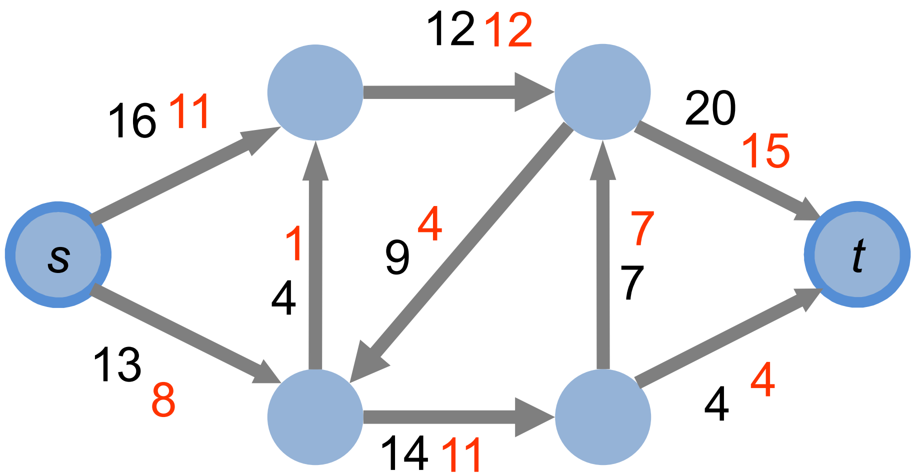
就是在residual network上面找到一個路徑，從 s 走到 t

- Given a network G and a flow f, an augmenting path p is a simple path from s to t in the residual network G_f.

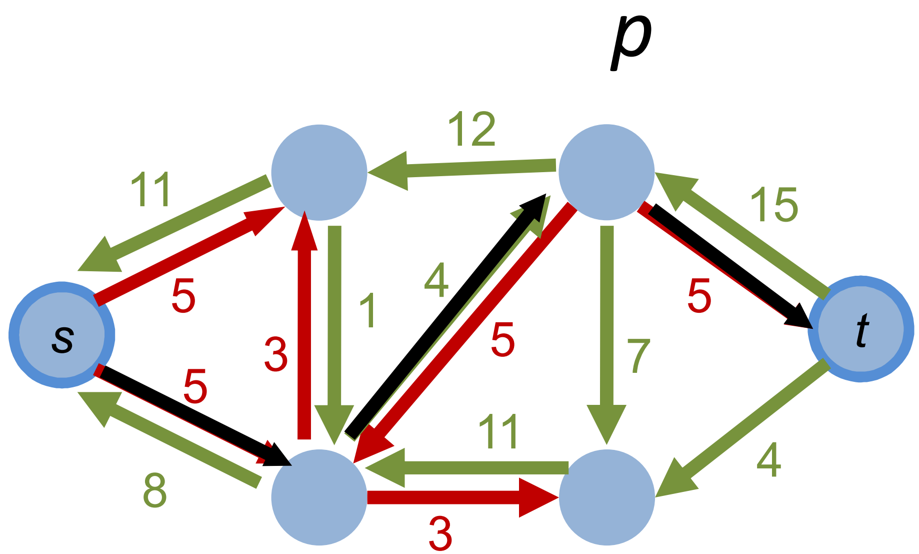


Augmenting path

- One more example:



G



G_f

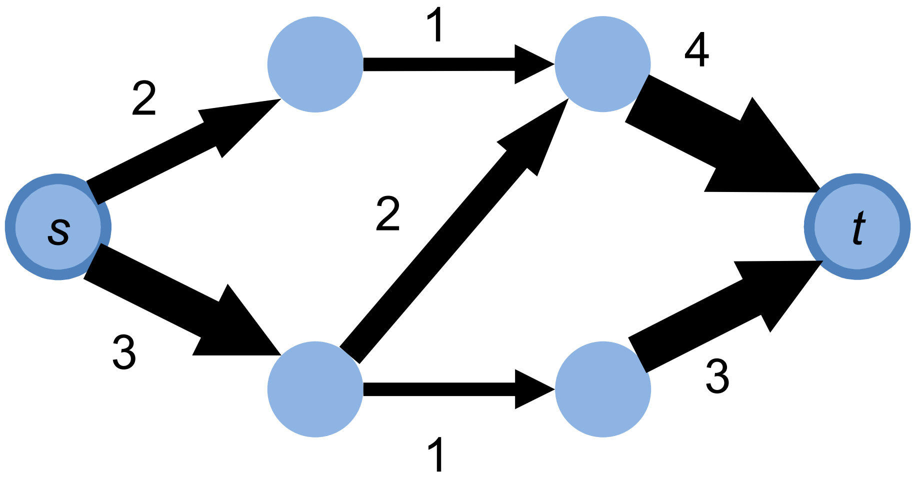
The Ford-Fulkerson method

FORD-FULKERSON-METHOD(G, s, t)

1. initialize flow f to 0
2. **while** there exists an augmenting path p in the residual network G_f, augment flow f along p
3. **return** f

Example

$f = 0$

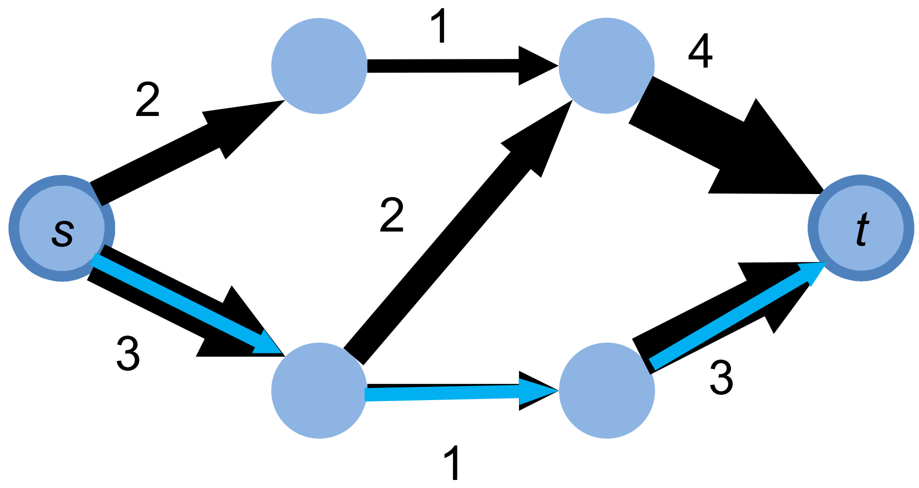


G

輸入圖

$f' = 1$

路徑中最小的capacity是 1，
流量不可以大於1（會爆），所以取1

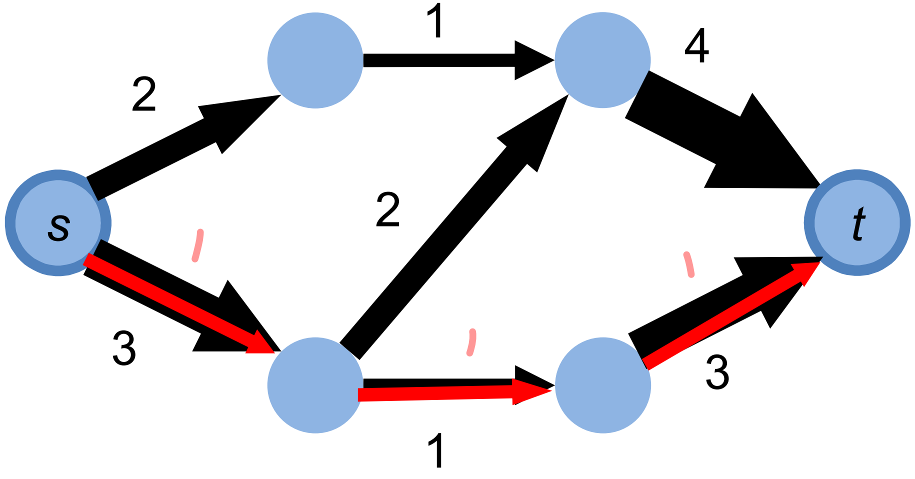


G_f

這個圖沒有原因，隨便選，從s走到t

Example

$f = 1$

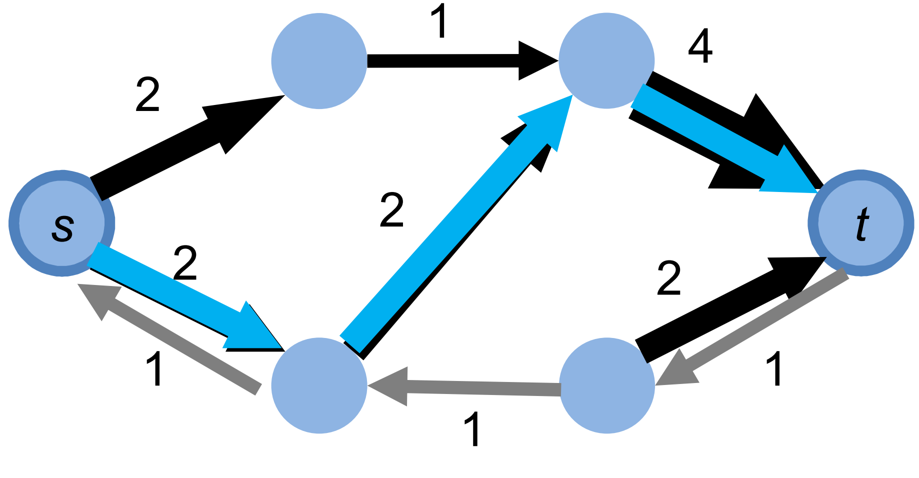


G

這個圖flow的流量是1

$f' = 2$

路徑中最小的capacity是 2，
流量不可以大於2（會爆），所以取2

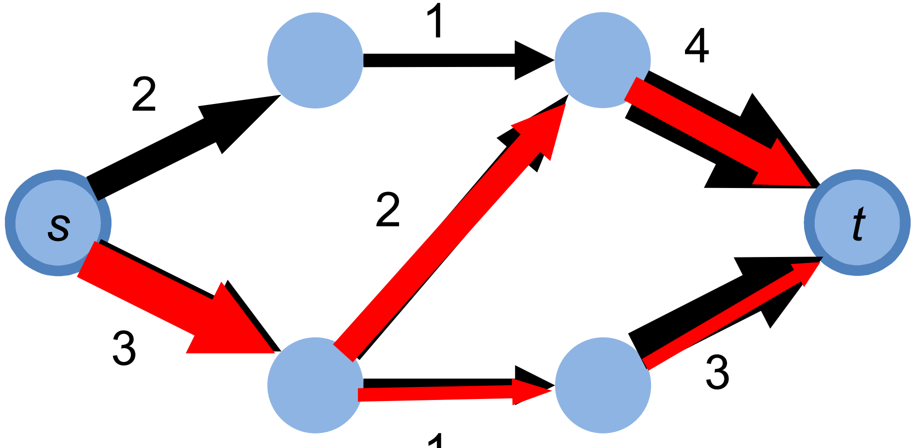


G_f

從capacity中扣掉用掉的流量，再飆上反向的邊

Example

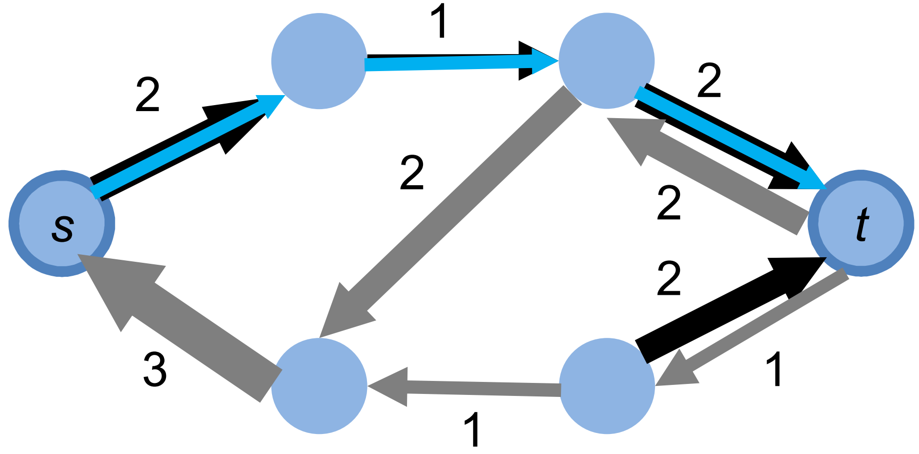
$f = 3$



G

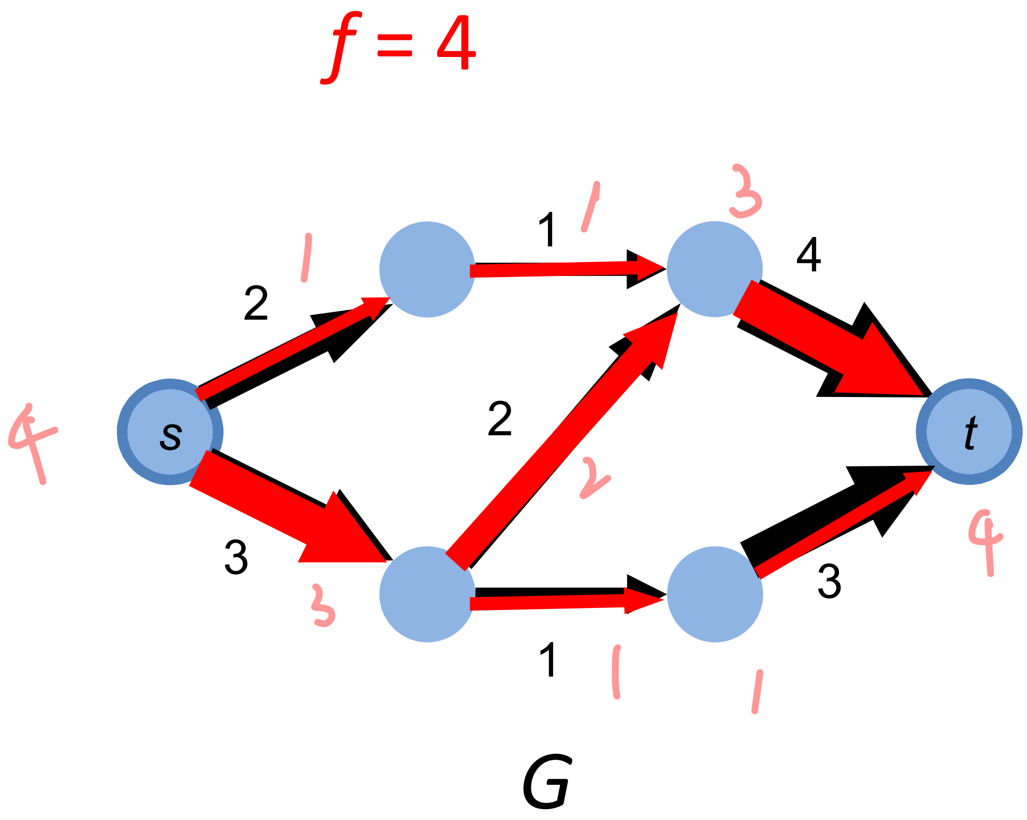
$f' = 1$

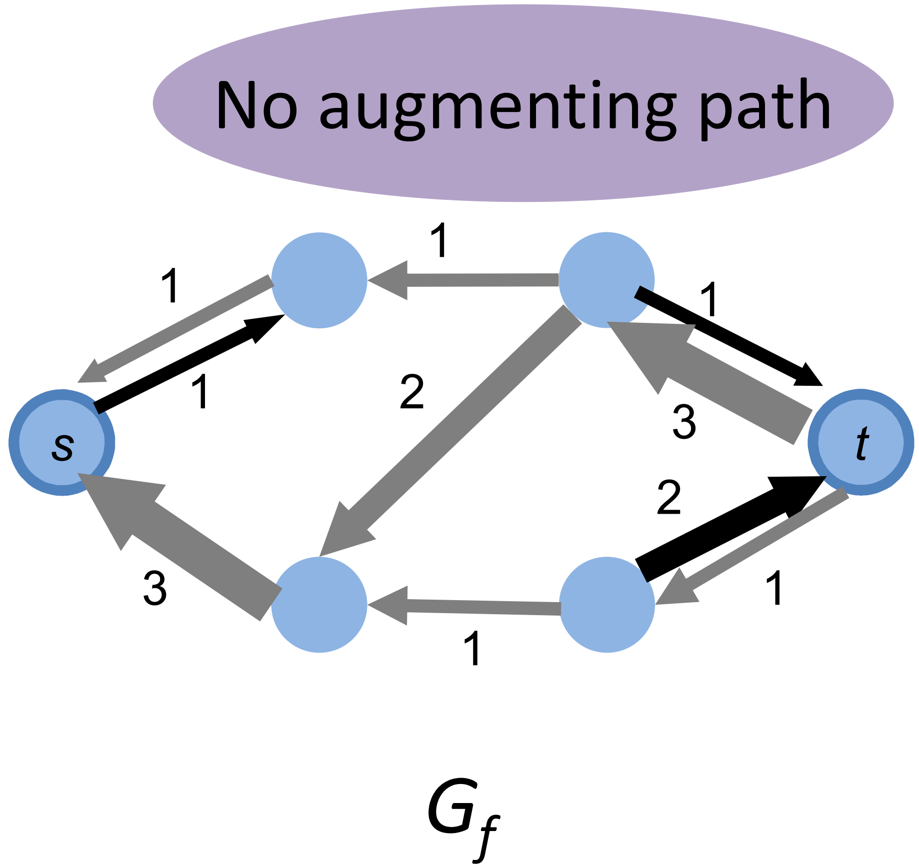
路徑中最小的capacity是 1，
流量不可以大於1（會爆），所以取1



G_f

Example



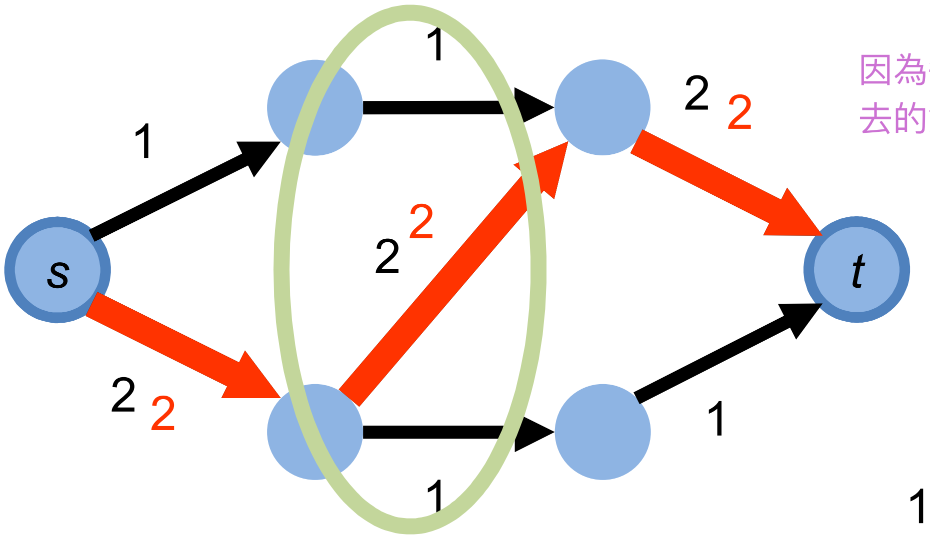


Since there is no augmenting path on G_f,
f reaches its maximal value.

Are “backward edges” needed?

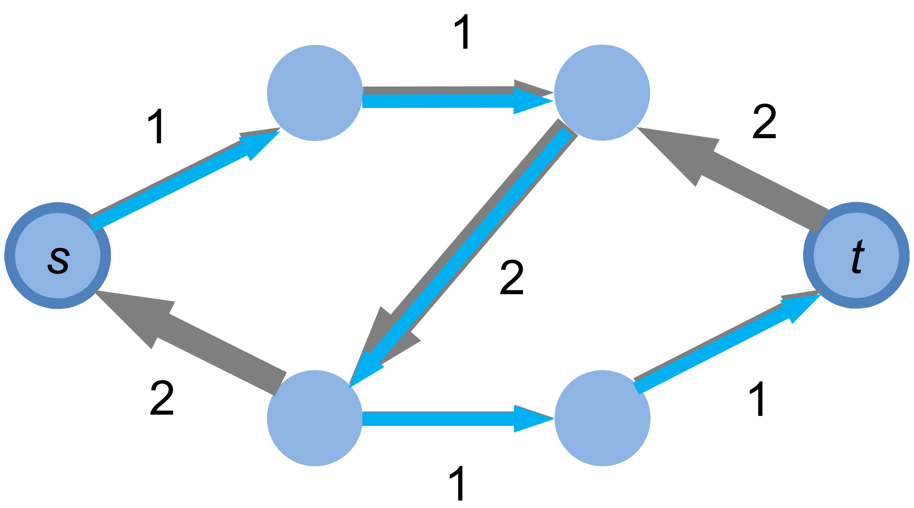
Q 為什麼要加上逆流的邊？是需要的嗎？為什麼不加會錯？

$f = 2$



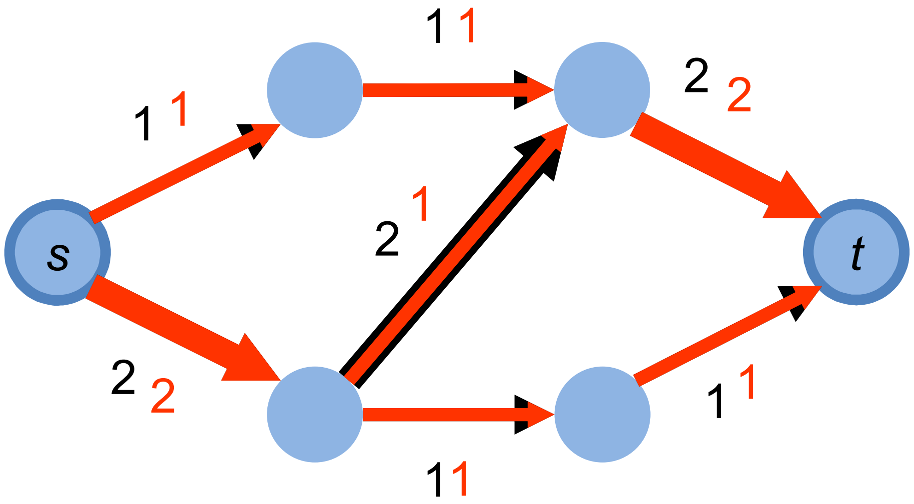
因為每次的augmenting path 是隨機選的，為了能夠修正過去的錯誤，所以加上backward edge。

有修補機制



The resulting flow

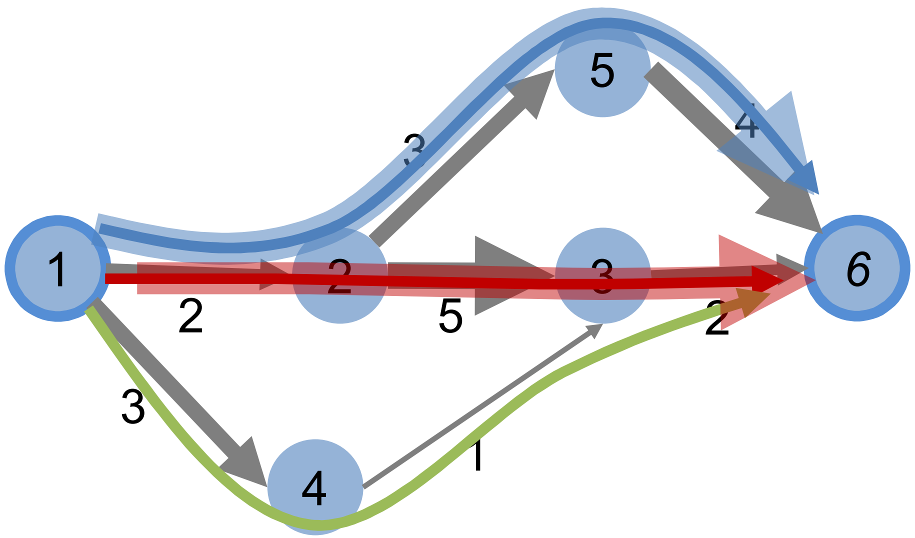
$f = 3$



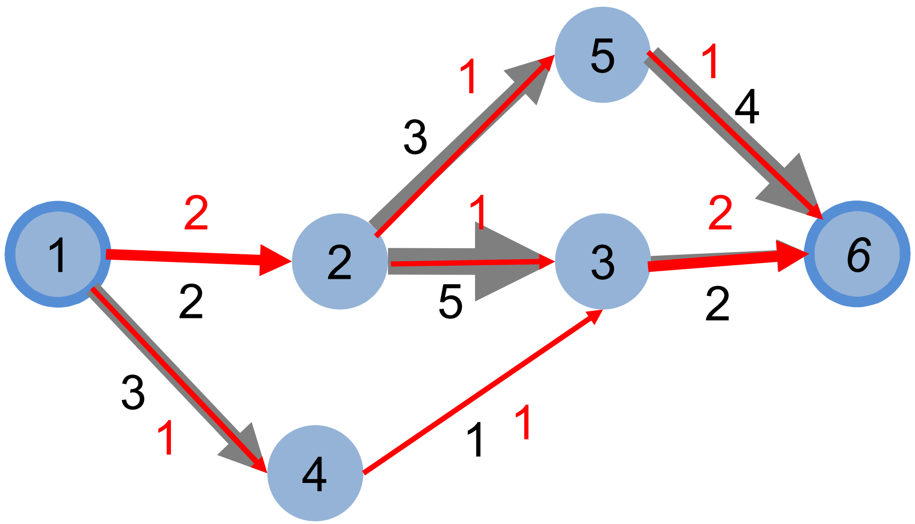
Exercise

流的方法、流的量不是唯一解，有很多種流法

$f_0 = 0$



Maximum flow? 3



起點流出2+1 = 3，
終點也流回去2+1 = 3

Residual network?

Augmenting paths?

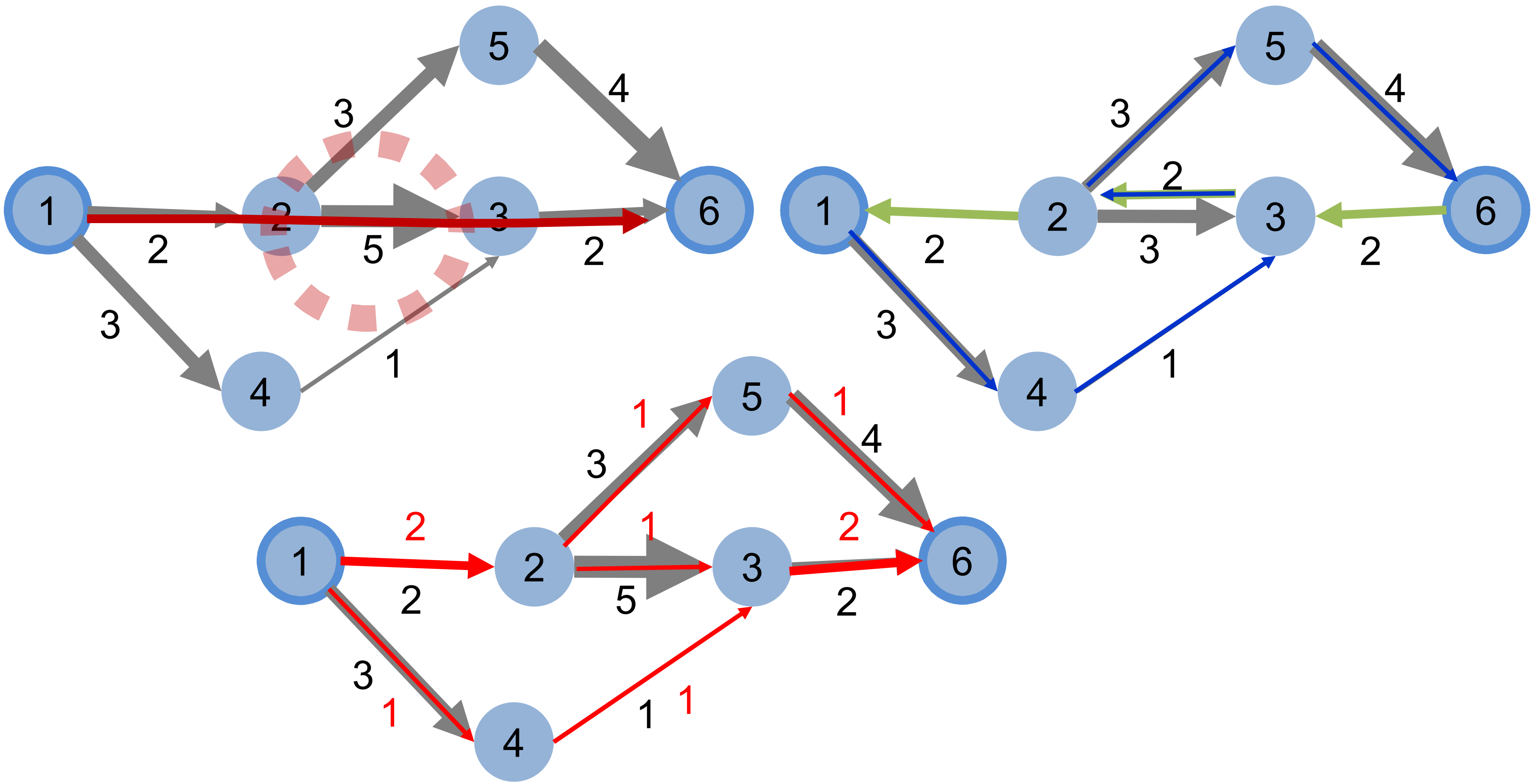
A: Augmenting path就是隨便選，只要能走到終點就行。
所以如果第一輪選了正中間那一條，但你沒有逆流機制，
你選到爛路你就回不來了，找不出最佳解

Exercise

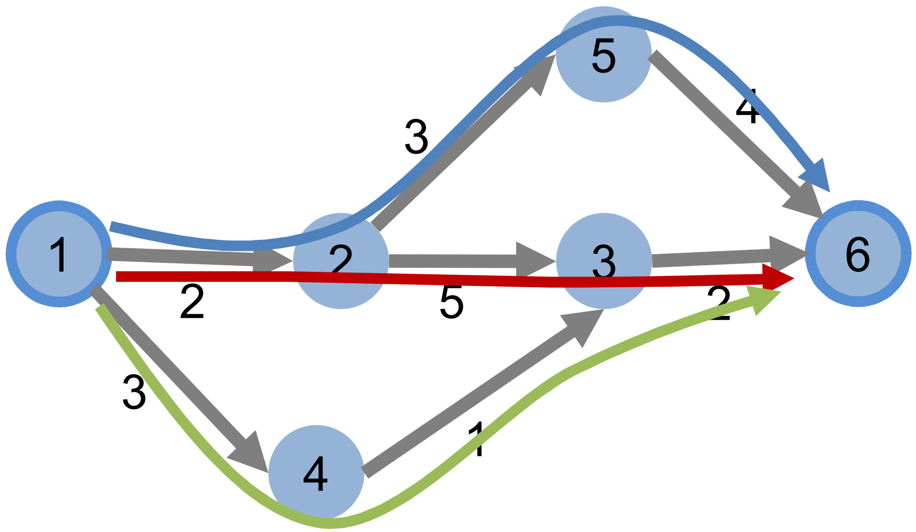
$f_1 = 2$

Residual network?

Augmenting paths?



- The final flow is maximal irrespective of a sequence of augmenting paths!



5/18 (日)
End