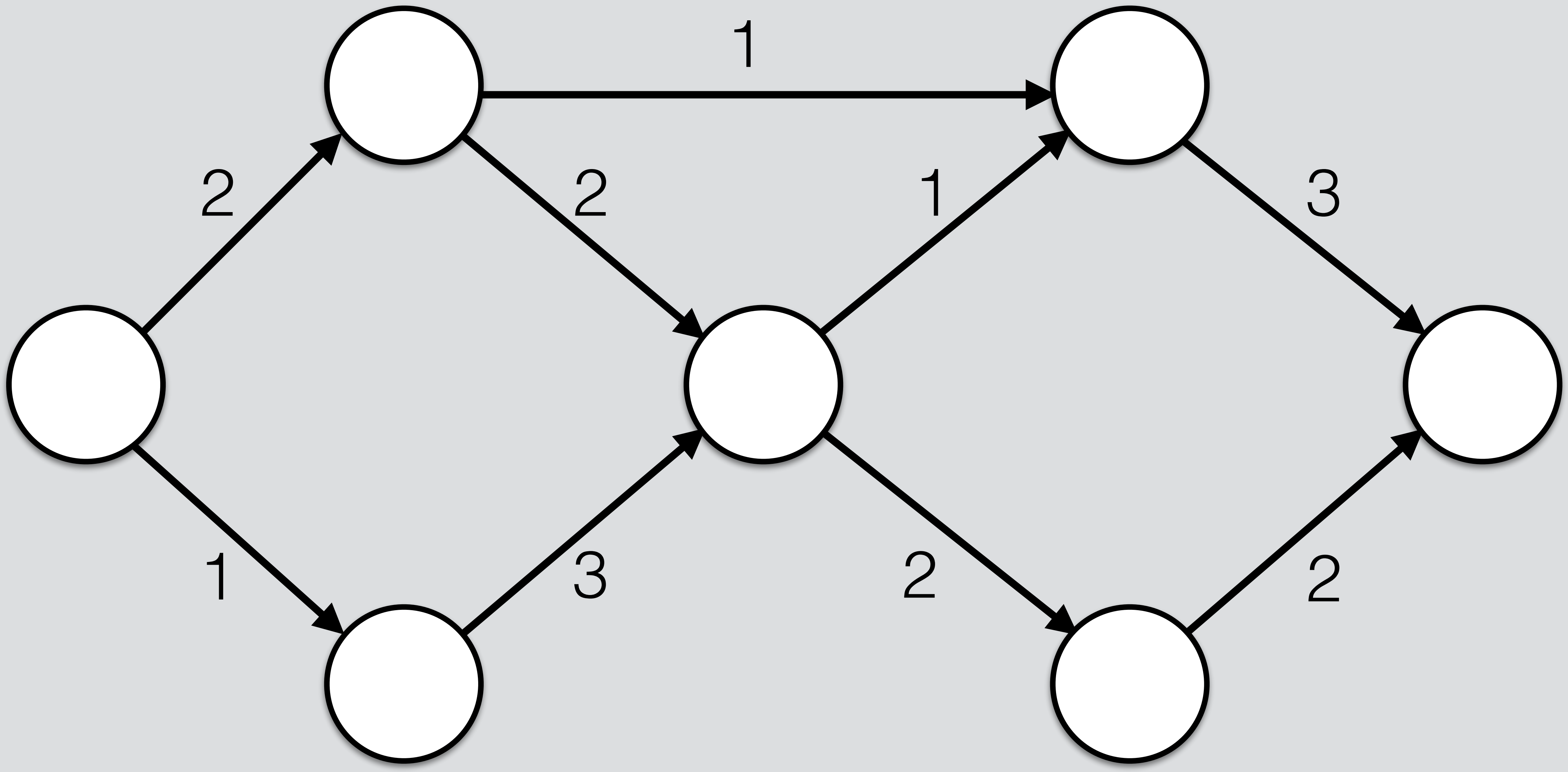
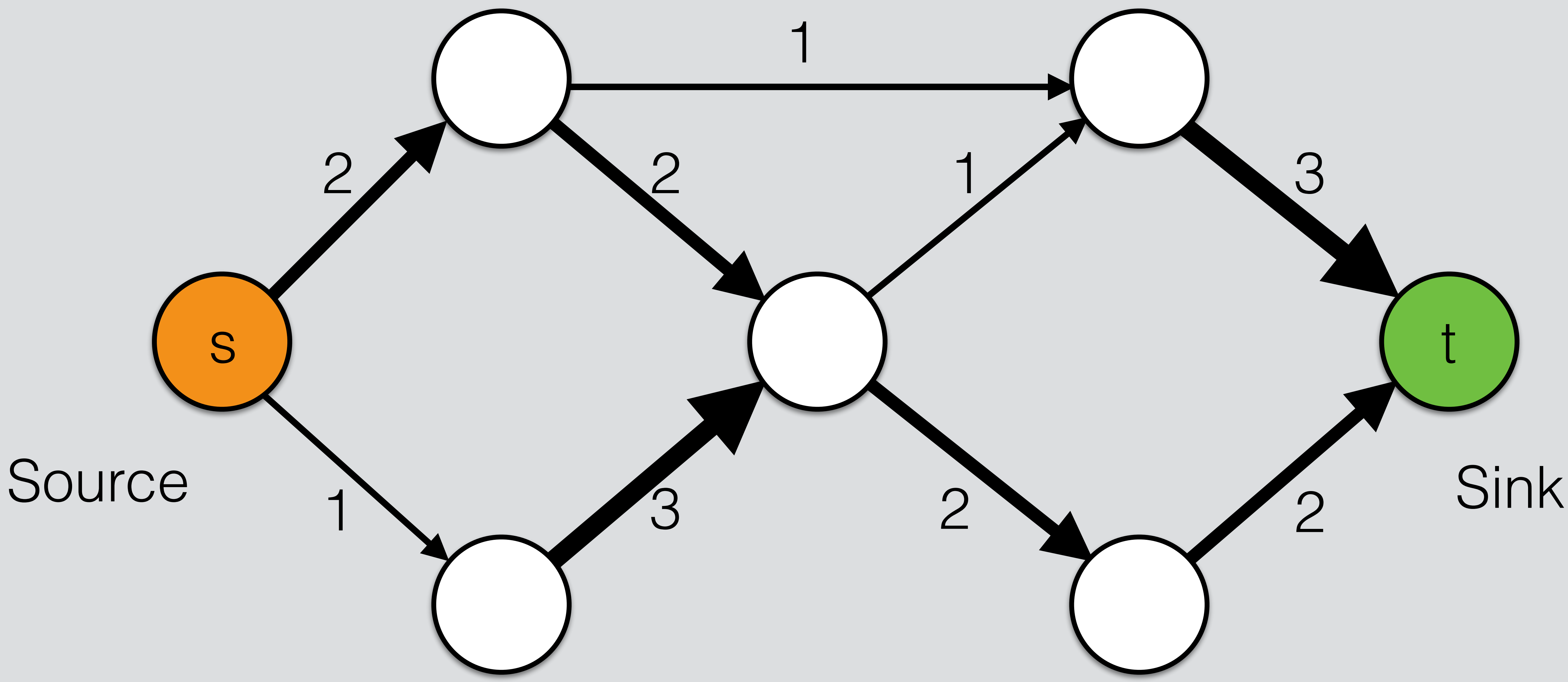
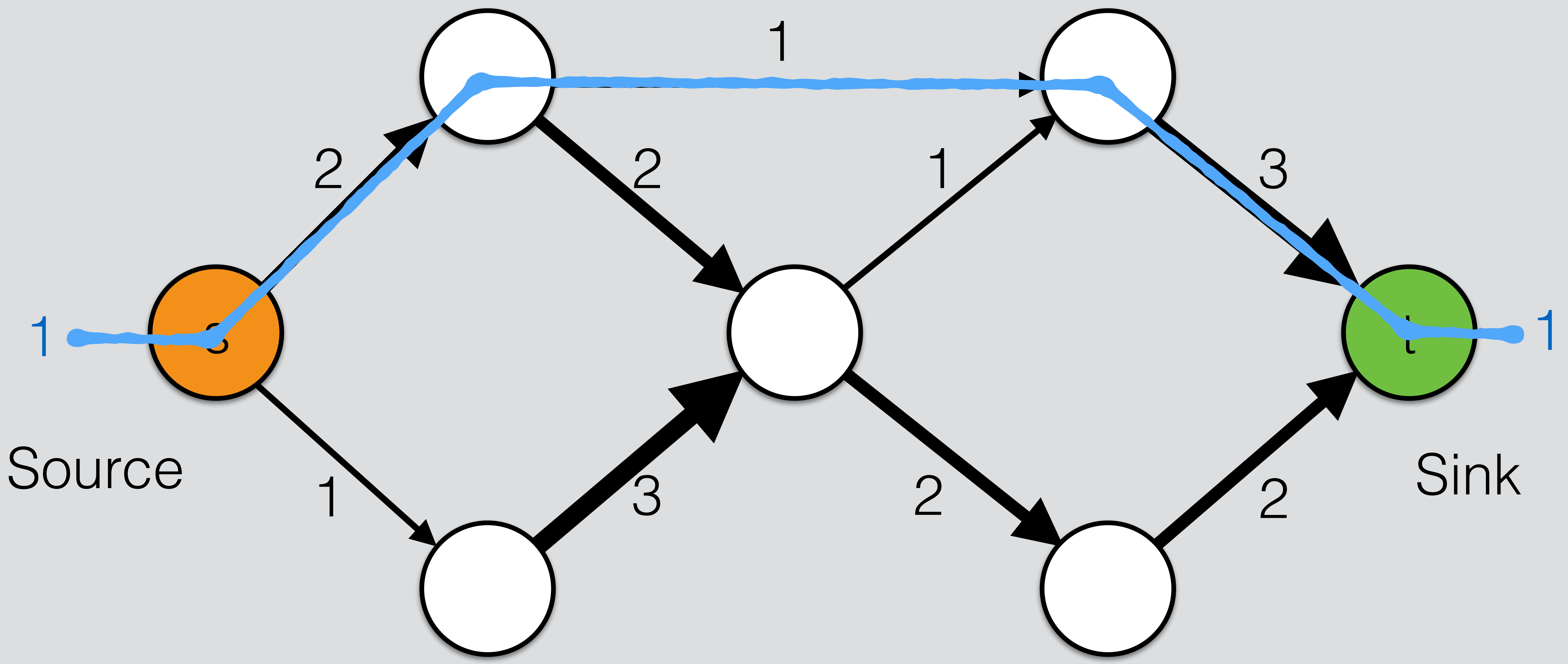
Flows


1
2
2
1
3
1
3
2
2

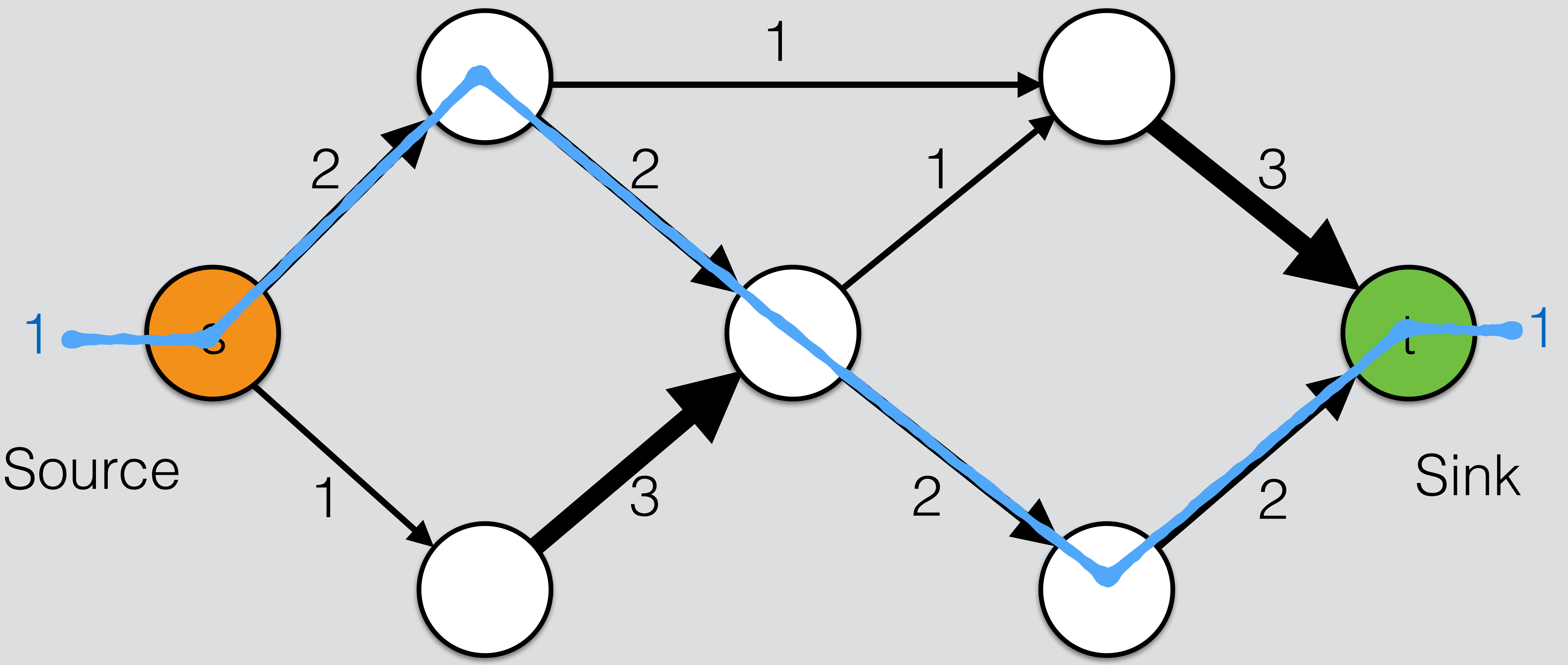


s
t
Source
Sink
1
2
2
1
3
1
3
2
2

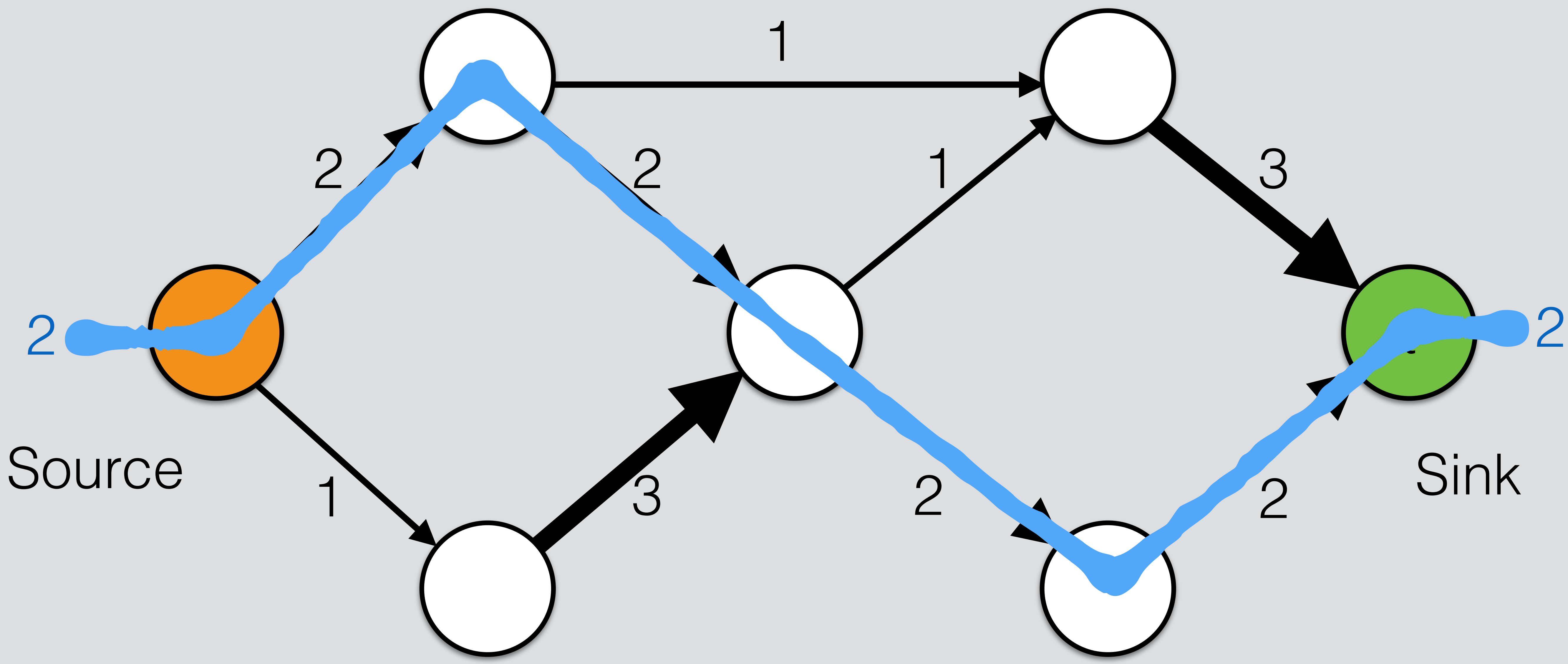
A *flow* of 1

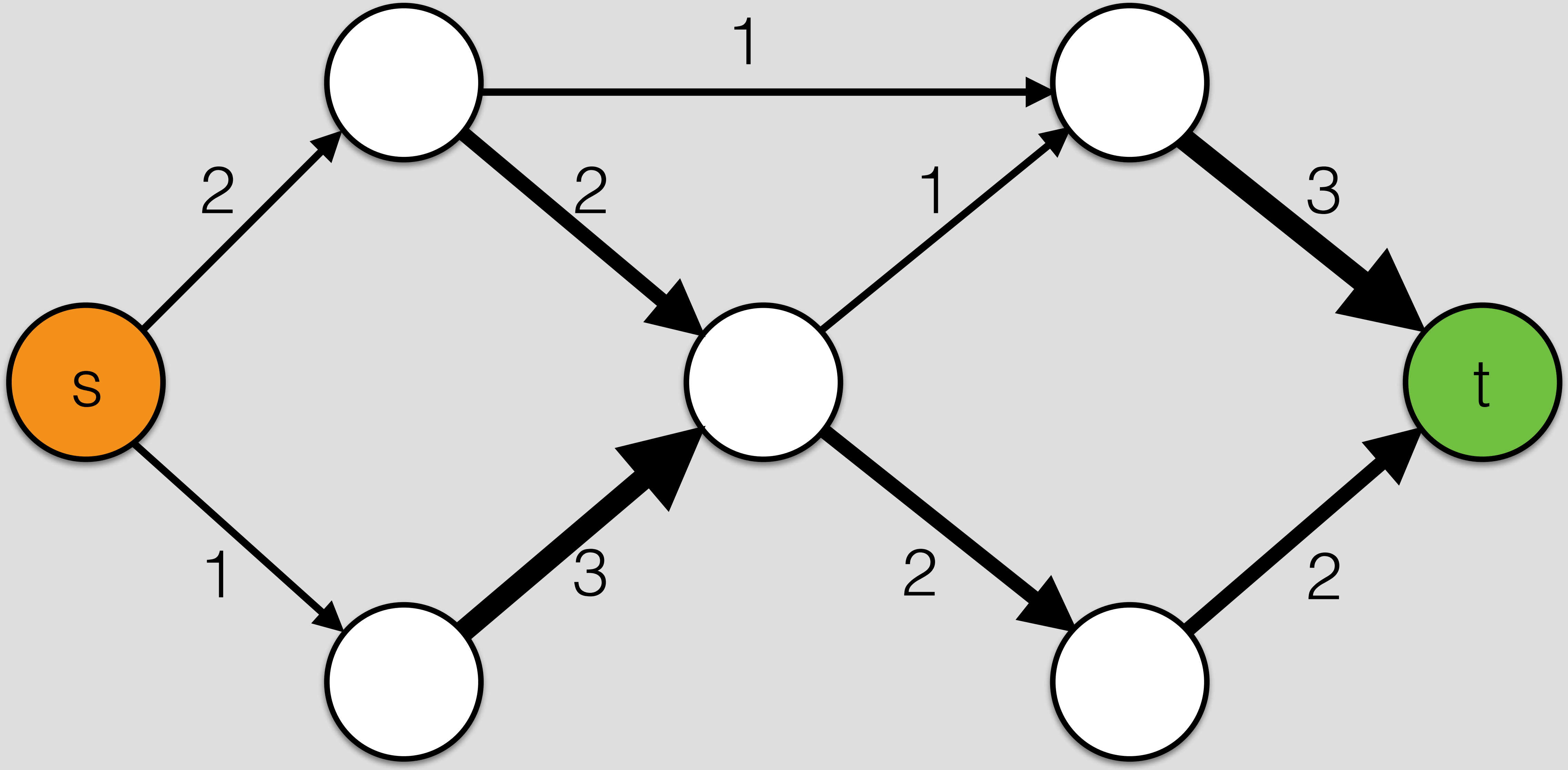


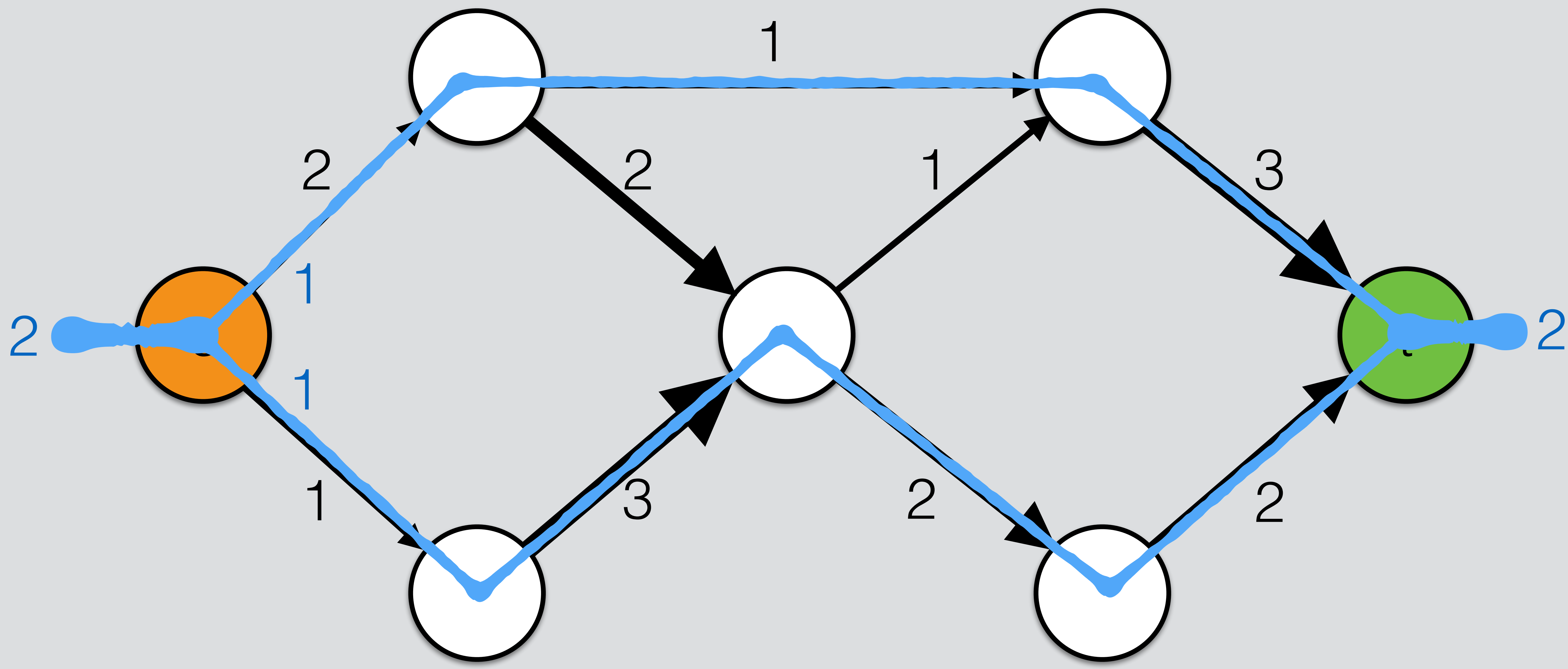
A *flow* of 1



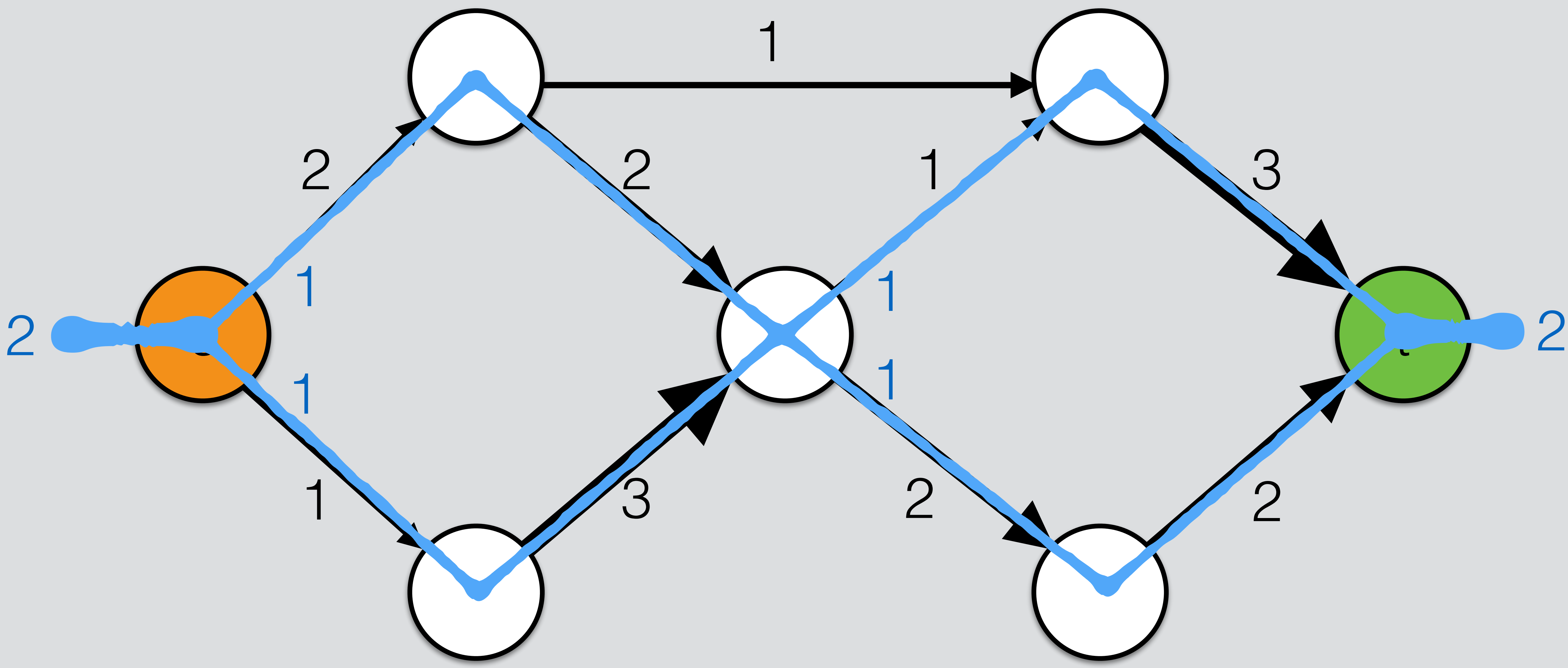
A *flow* of 2



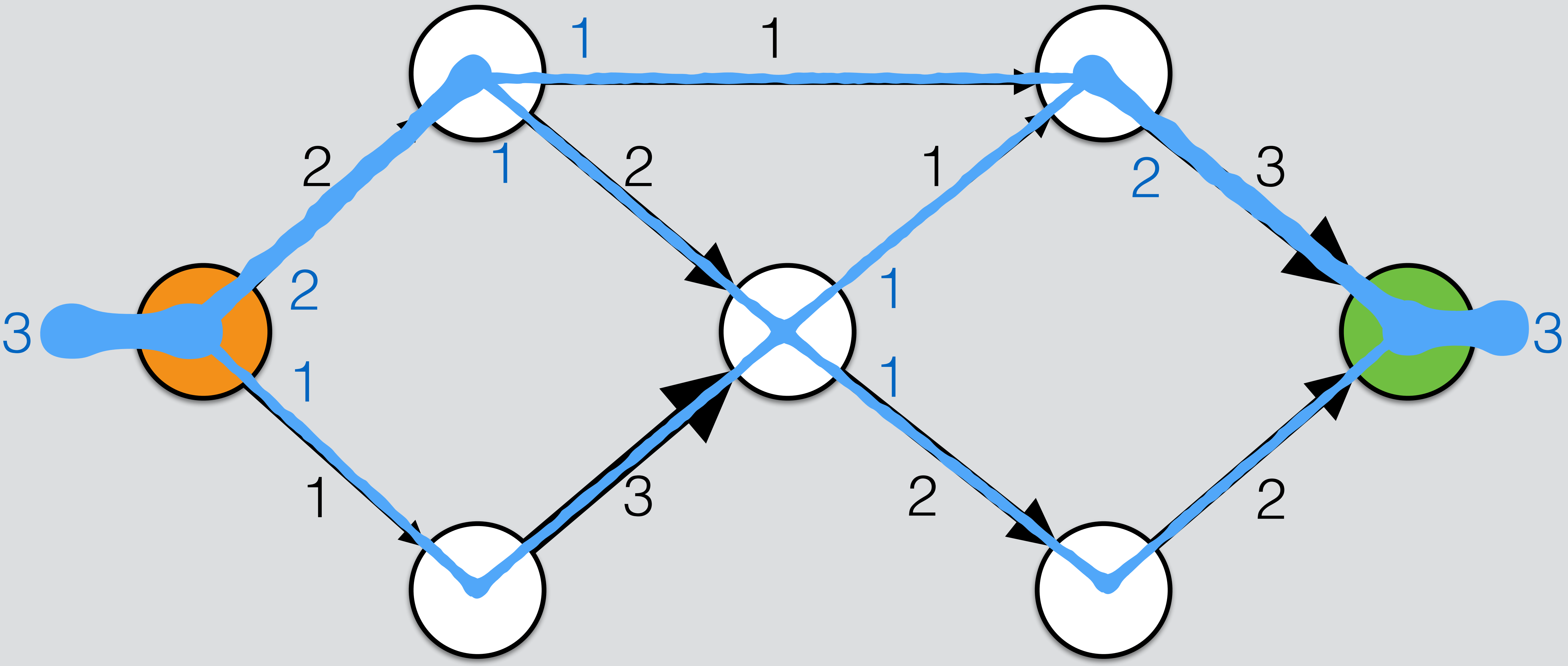

s
t
2
1
1
2
3
1
3
2
2



2
1
1
2
1
2
1
3
3
1
2
2
2

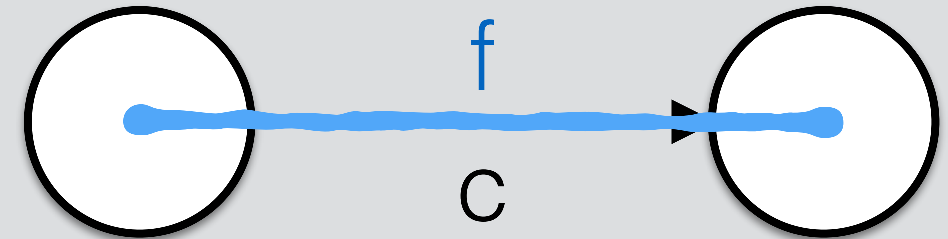


2
1
1
2
1
2
1
1
1
3
3
2
2
2


1
1
2
1
2
1
3
2
3
2
1
3
1
2
1
3
2
3
2

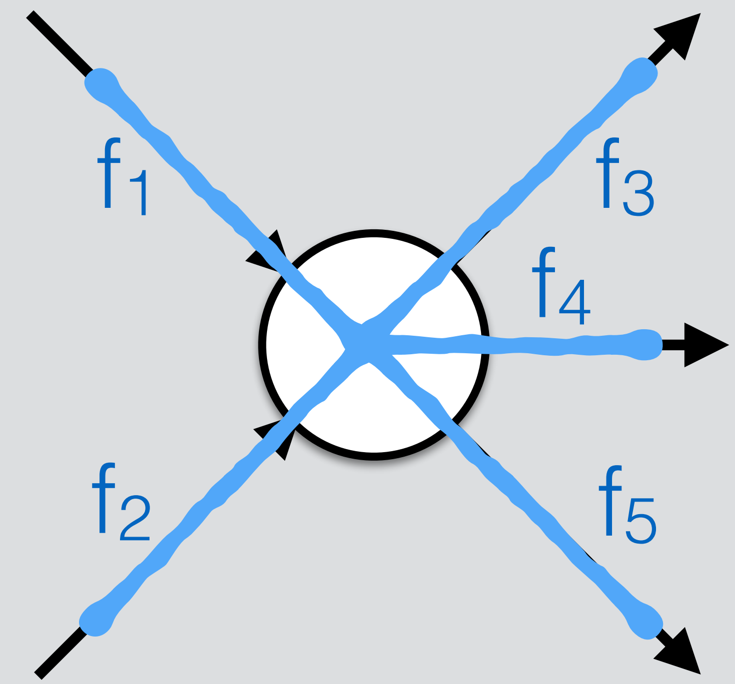
Rules of flows

For each edge, flow on edge is less than edge's capacity



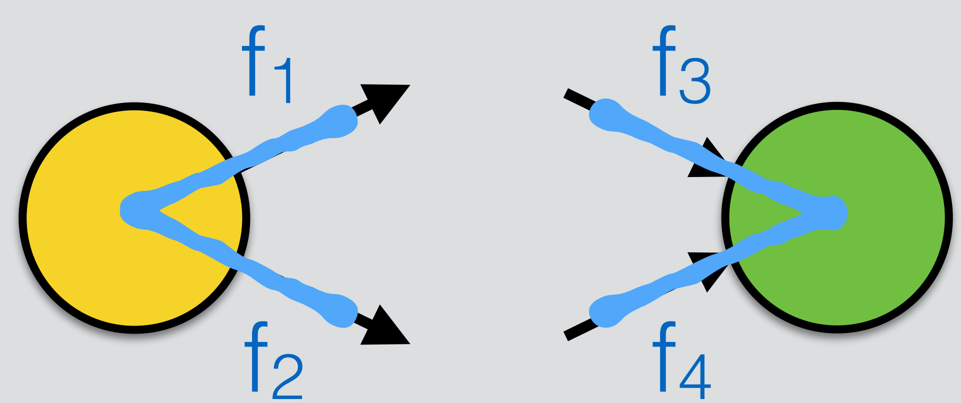
$f \leq c$

For non-source/sink vertices, in-flow and out-flow of node are equal.

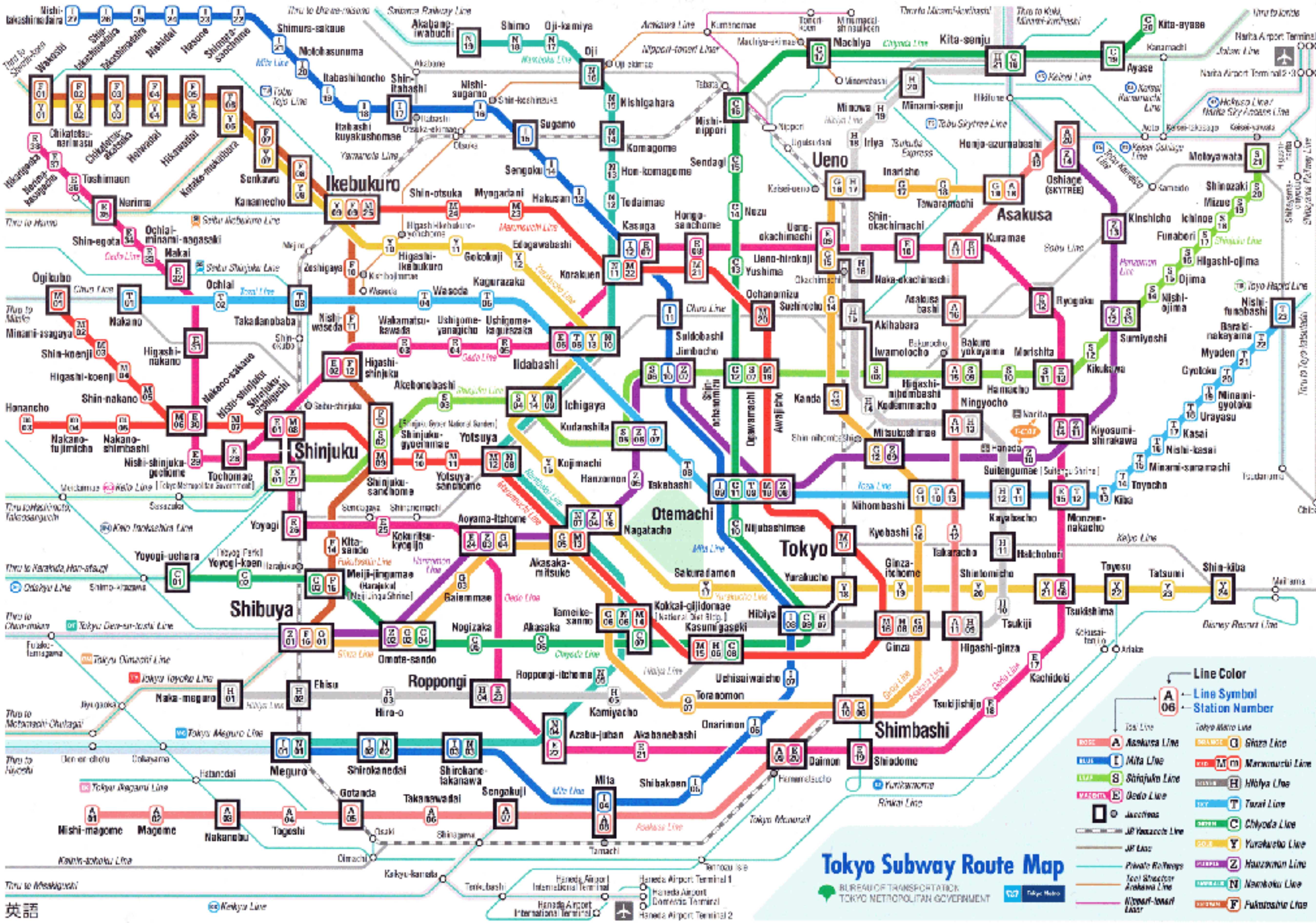


$f_1 + f_2 = f_3 + f_4 + f_5$

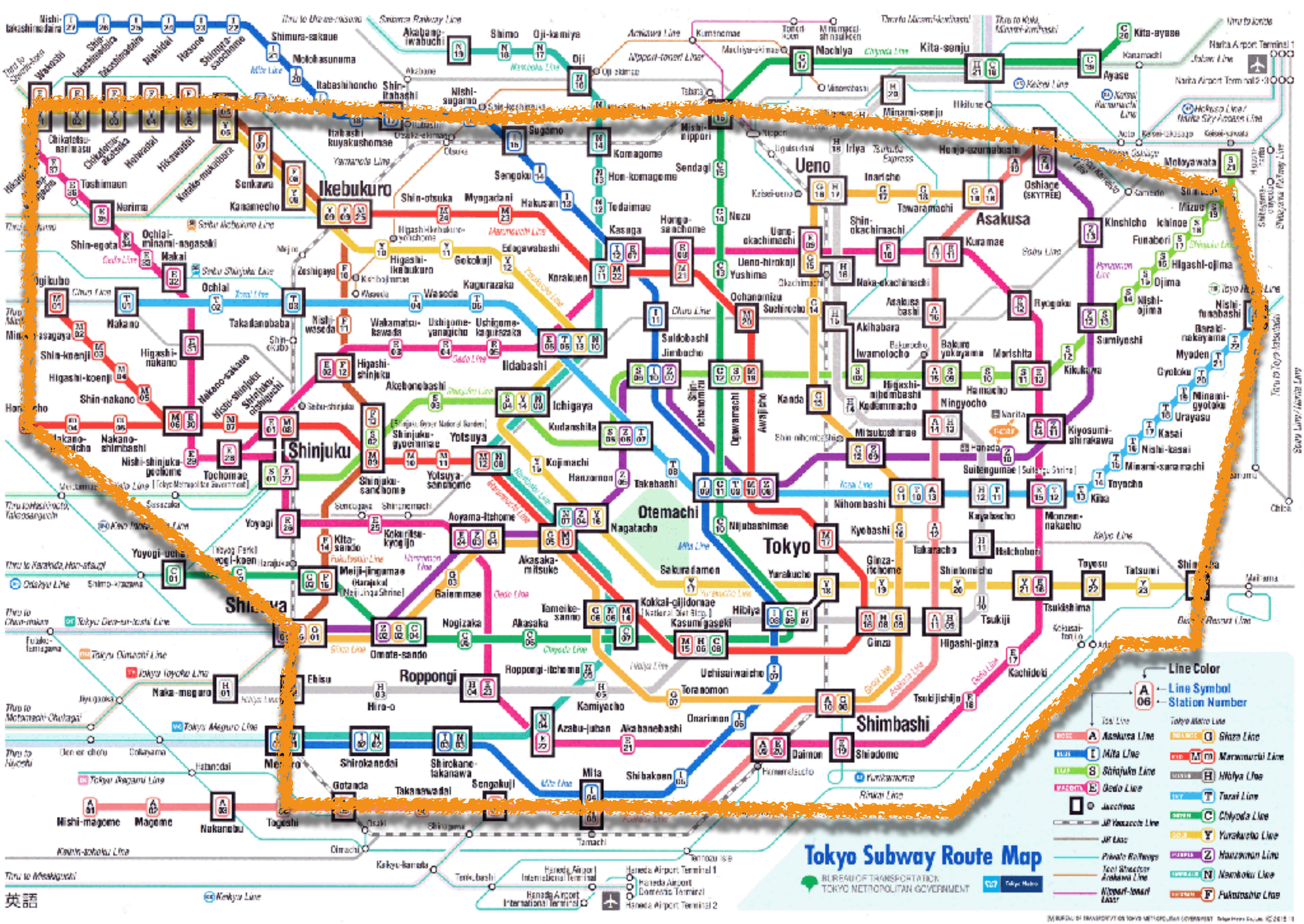
Out-flow of source and in-flow of sink are equal.



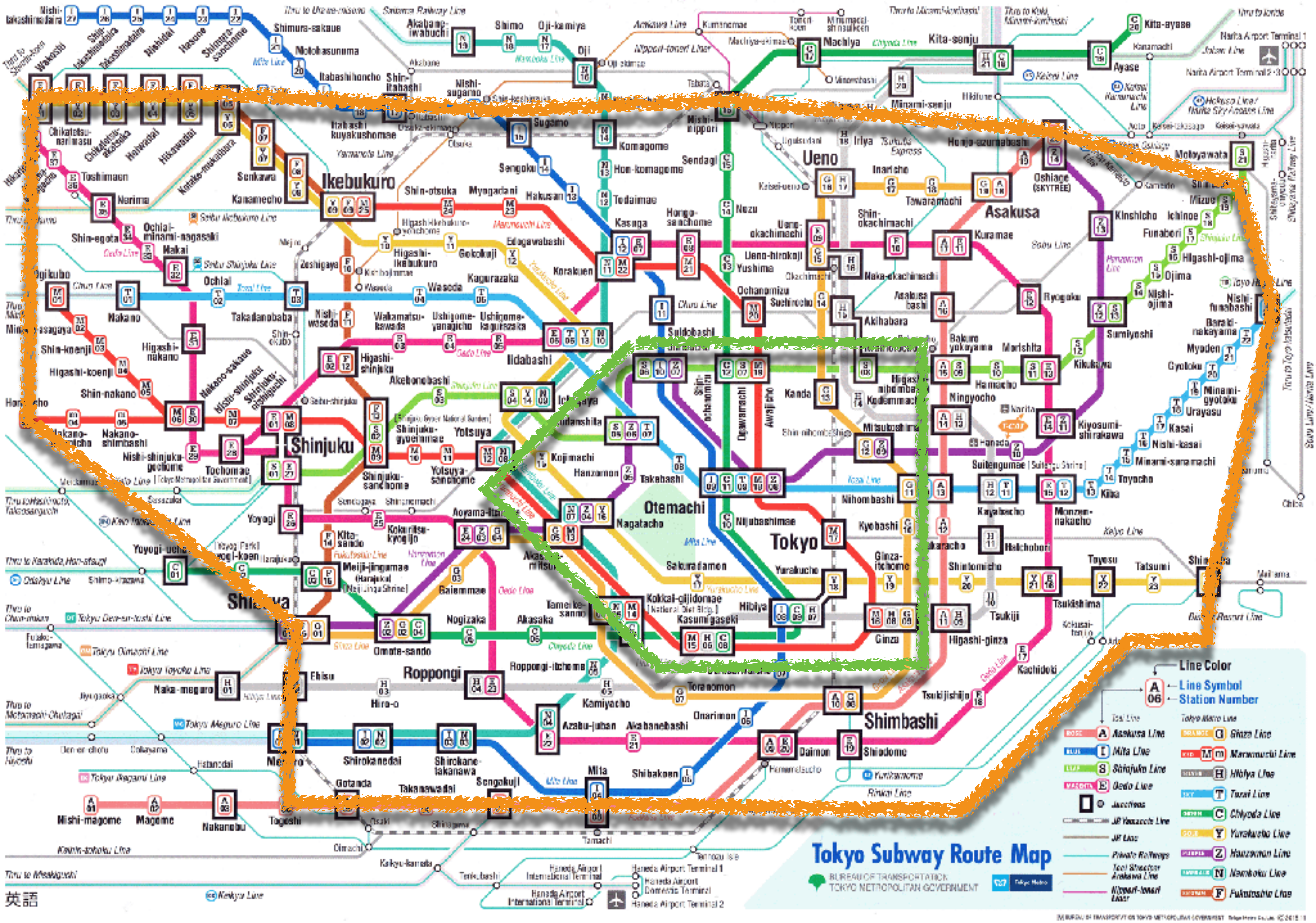
$f_1 + f_2 = f_3 + f_4$



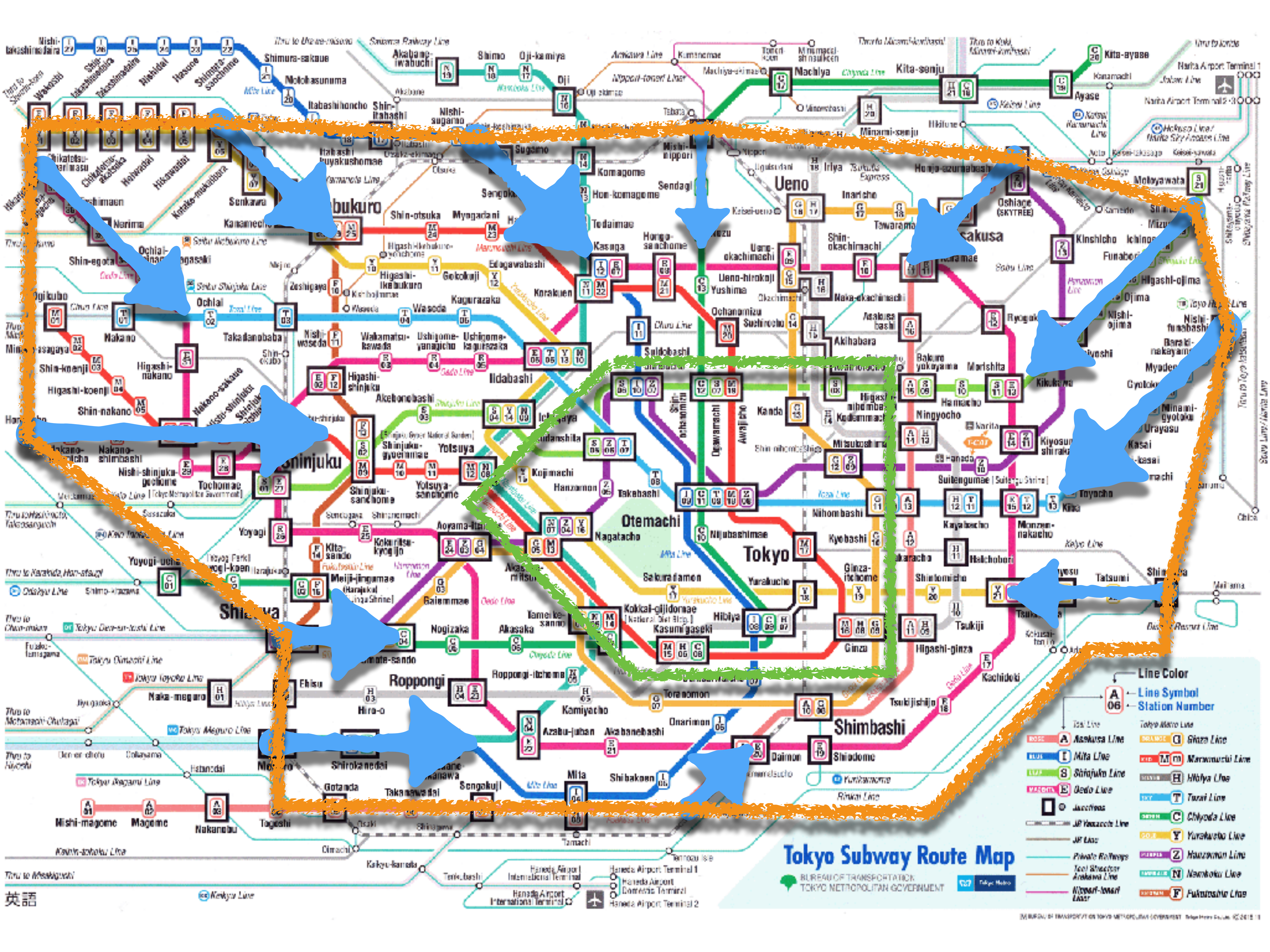
Tokyo Subway Route Map
BUREAU OF TRANSPORTATION
TOKYO METROPOLITAN GOVERNMENT
Line Color
Line Symbol
Station Number
Asakusa Line
Mita Line
Shinjuku Line
Oedo Line
Ginza Line
Marunouchi Line
Hibiya Line
Tozai Line
Chiyoda Line
Yurakucho Line
Hanzomon Line
Namboku Line
Fukutoshin Line
Ikebukuro
Shinjuku
Shibuya
Ueno
Asakusa
Otemachi
Tokyo
Shimbashi
Roppongi
Meguro
Nishi-takashimadaira
Shimura-sakaue
Itabashihoncho
Sugamo
Kasuga
Korakuen
Iidabashi
Ichigaya
Kudanshita
Jimbocho
Akihabara
Ningyocho
Kiyosumi-shirakawa
Sumiyoshi
Kikukawa
Oshiage (SKYTREE)
Kuramae
Ryogoku
Monzen-nakacho
Toyosu
Shin-kiba
Tsukishima
Kachidoki
Tsukiji
Higashi-ginza
Ginza
Hibiya
Yurakucho
Kasumigaseki
Toranomon
Akasaka
Nogizaka
Omote-sando
Gaiemmae
Aoyama-itchome
Yotsuya
Shinjuku-sanchome
Nakano
Ogikubo
Nerima
Hikarigaoka
Nakanobu
Gotanda
Sengakuji
Mita
Shibakoen
Daimon
Shiodome
Tsukijishijo
Nishi-magome
Magome
Togoshi
Takanawadai
Shirokanedai
Shirokane-takanawa
Azabu-juban
Ebisu
Naka-meguro
Hiro-o
Kamiyacho
Onarimon
Uchisaiwaicho
英語

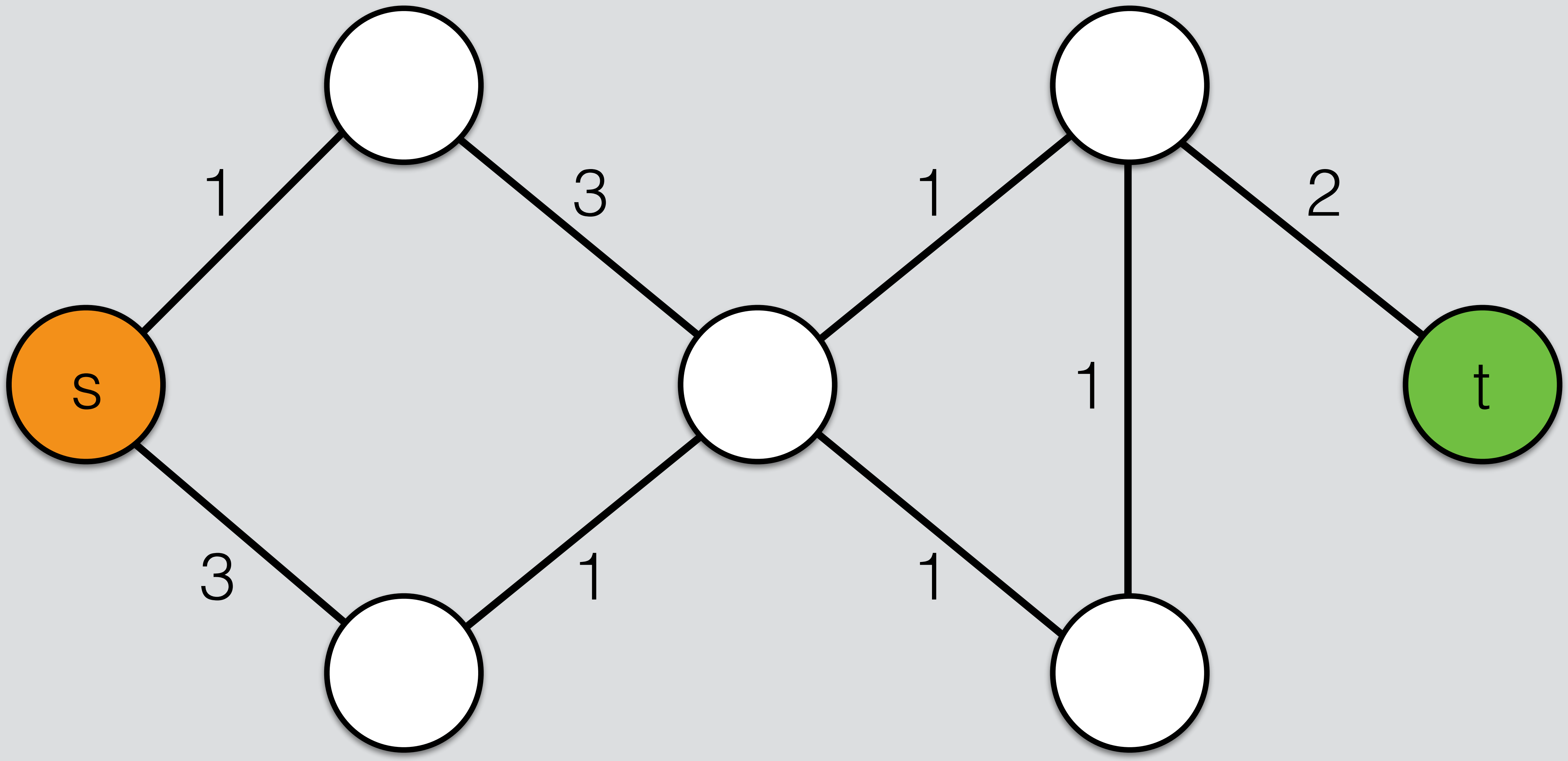


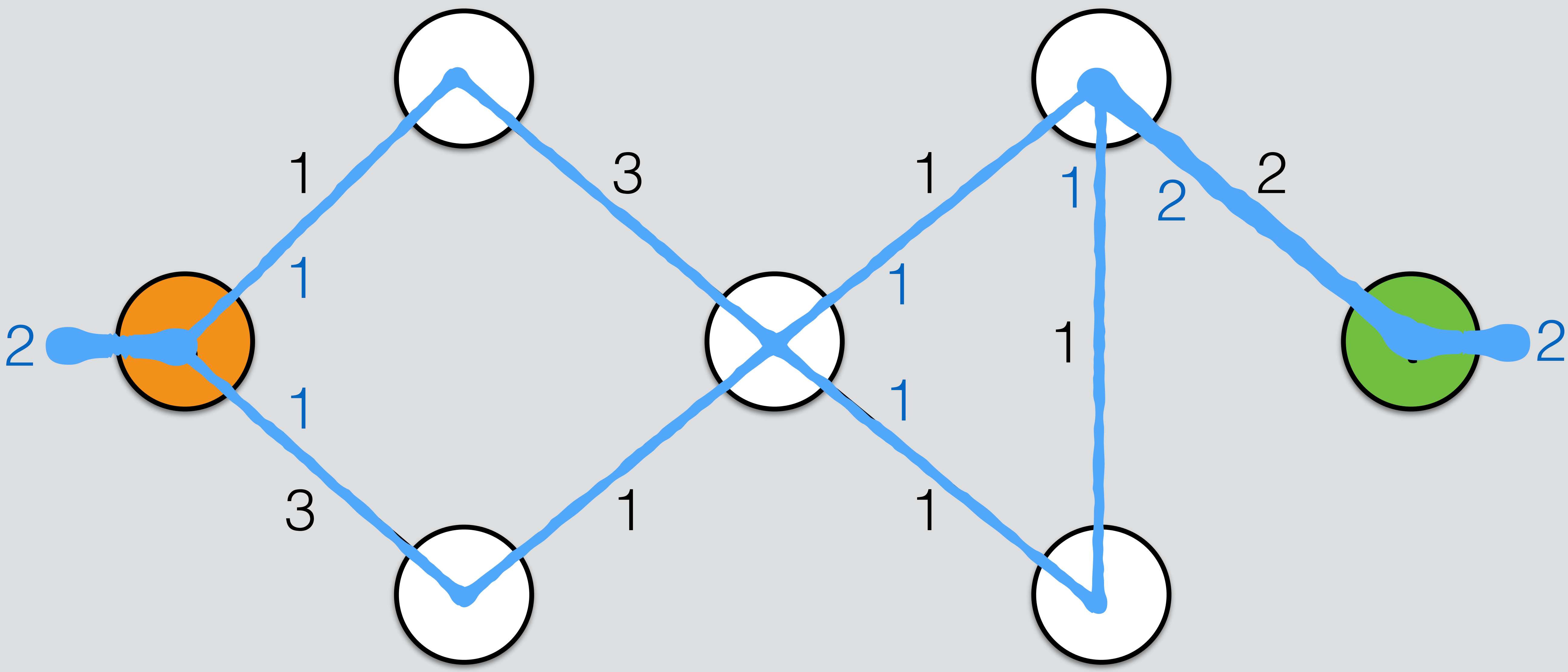
Tokyo Subway Route Map
BUREAU OF TRANSPORTATION
TOKYO METROPOLITAN GOVERNMENT
Line Color
Line Symbol
Station Number
Asakusa Line
Mita Line
Shinjuku Line
Oedo Line
Ginza Line
Marunouchi Line
Hibiya Line
Tozai Line
Chiyoda Line
Yurakucho Line
Hanzomon Line
Namboku Line
Fukutoshin Line
Ikebukuro
Shinjuku
Shibuya
Ueno
Asakusa
Tokyo
Otemachi
Shimbashi
Roppongi
Iidabashi
Ichigaya
Yotsuya
Akihabara
Ryogoku
Nihombashi
Kasumigaseki
Meguro
Ebisu
Gotanda
Shirokanedai
Mita
Daimon
Shiodome
Kita-senju
Oshiage (SKYTREE)
Ochanomizu
Kanda
Kudanshita
Akasaka
Nagatacho
Hibiya
Ginza
Tsukiji
Tsukishima
Toyosu
Kiyosumi-shirakawa
Monzen-nakacho
Nakano
Ogikubo
Nerima
Sugamo
Nishi-nippori
Komagome
Nezu
Korakuen
Kasuga
Suidobashi
Jimbocho
Takadanobaba
Waseda
Omote-sando
Nogizaka
Aoyama-itchome
Azabu-juban
Hiro-o
Tameike-sanno
Toranomon
Onarimon
Shibakoen
Sengakuji
Kachidoki
Nishi-magome
Magome
Nakanobu
Togoshi
Haneda Airport Terminal 1
Haneda Airport Terminal 2
Narita Airport Terminal 1
英語



Tokyo Subway Route Map
BUREAU OF TRANSPORTATION
TOKYO METROPOLITAN GOVERNMENT
Line Color
Line Symbol
Station Number
Asakusa Line
Mita Line
Shinjuku Line
Oedo Line
Ginza Line
Marunouchi Line
Hibiya Line
Tozai Line
Chiyoda Line
Yurakucho Line
Hanzomon Line
Namboku Line
Fukutoshin Line
Ikebukuro
Shinjuku
Shibuya
Ueno
Asakusa
Tokyo
Otemachi
Shimbashi
Roppongi
Ginza
Nishi-takashimadaira
Shimura-sakaue
Itabashihoncho
Sugamo
Komagome
Hon-komagome
Todaimae
Nishi-nippori
Nezu
Sendagi
Machiya
Kita-senju
Minami-senju
Kita-ayase
Ayase
Iriya
Inaricho
Tawaramachi
Honjo-azumabashi
Oshiage (SKYTREE)
Kuramae
Kasuga
Korakuen
Hongo-sanchome
Ueno-okachimachi
Ueno-hirokoji
Yushima
Naka-okachimachi
Ochanomizu
Suehirocho
Akihabara
Ryogoku
Kinshicho
Sumiyoshi
Morishita
Kikukawa
Ojima
Higashi-ojima
Funabori
Ichinoe
Mizue
Ichikawa
Nishi-ojima
Nishi-funabashi
Baraki-nakayama
Myoden
Gyotoku
Minami-gyotoku
Urayasu
Kasai
Nishi-kasai
Minami-sunamachi
Toyocho
Kiba
Monzen-nakacho
Kiyosumi-shirakawa
Suitengumae
Ningyocho
Hamacho
Kayabacho
Hatchobori
Kyobashi
Nihombashi
Mitsukoshimae
Kanda
Jimbocho
Suidobashi
Iidabashi
Kagurazaka
Waseda
Edogawabashi
Gokokuji
Myogadani
Shin-otsuka
Higashi-ikebukuro
Senkawa
Kanamecho
Hikawadai
Heiwadai
Chikatetsu-narimasu
Toshimaen
Nerima
Shin-egota
Ochiai-minami-nagasaki
Ochiai
Nakano
Ogikubo
Takadanobaba
Shin-koenji
Higashi-koenji
Shin-nakano
Nakano-sakaue
Nishi-shinjuku
Tochomae
Shinjuku-gyoemmae
Shinjuku-sanchome
Yotsuya
Yotsuya-sanchome
Ichigaya
Kudanshita
Kojimachi
Hanzomon
Takebashi
Nagatacho
Akasaka
Akasaka-mitsuke
Kokkai-gijidomae
Kasumigaseki
Hibiya
Yurakucho
Sakuradamon
Nijubashimae
Ginza-itchome
Higashi-ginza
Tsukiji
Shintomicho
Tsukishima
Kachidoki
Toyosu
Tatsumi
Tsukijishijo
Shiodome
Daimon
Onarimon
Shibakoen
Mita
Akabanebashi
Azabu-juban
Kamiyacho
Toranomon
Roppongi-itchome
Nogizaka
Omote-sando
Gaiemmae
Meiji-jingumae
Kokuritsu-kyogijo
Aoyama-itchome
Yoyogi
Yoyogi-uehara
Ebisu
Hiro-o
Naka-meguro
Meguro
Shirokanedai
Shirokane-takanawa
Sengakuji
Takanawadai
Gotanda
Togoshi
Nakanobu
Magome
Nishi-magome
英語


Tokyo Subway Route Map
BUREAU OF TRANSPORTATION
TOKYO METROPOLITAN GOVERNMENT
Line Color
Line Symbol
Station Number
Asakusa Line
Mita Line
Shinjuku Line
Oedo Line
Ginza Line
Marunouchi Line
Hibiya Line
Tozai Line
Chiyoda Line
Yurakucho Line
Hanzomon Line
Namboku Line
Fukutoshin Line
Ikebukuro
Ueno
Asakusa
Shinjuku
Tokyo
Otemachi
Shibuya
Roppongi
Shimbashi
Ginza
Yotsuya
Ichigaya
Iidabashi
Kasuga
Akihabara
Ochanomizu
Kanda
Nihombashi
Hibiya
Kasumigaseki
Akasaka
Omote-sando
Ebisu
Meguro
Gotanda
Shinagawa
Nakano
Ogikubo
Kitasenju
Ayase
Kita-ayase
Oshiage (SKYTREE)
Ryogoku
Monzen-nakacho
Toyocho
Nishi-magome
Magome
Nakanobu
Togoshi
Tsukiji
Tsukishima
Kachidoki
Daimon
Shiodome
Mita
Shirokanedai
Azabu-juban
Akabanebashi
Onarimon
Toranomon
Kamiyacho
Nogizaka
Aoyama-itchome
Meiji-jingumae
Kokkai-gijidomae
Sakuradamon
Yurakucho
Kyobashi
Kayabacho
Hatchobori
Suitengumae
Ningyocho
Kodemmacho
Mitsukoshimae
Kinshicho
Ojima
Higashi-ojima
Funabori
Ichinoe
Mizue
Shinozaki
Morishita
Kikukawa
Sumiyoshi
Nishi-ojima
Tawaramachi
Inaricho
Ueno-hirokoji
Naka-okachimachi
Yushima
Nezu
Sendagi
Todaimae
Komagome
Hon-komagome
Sugamo
Nishi-sugamo
Oji
Shimo
Akabane-iwabuchi
Shimura-sakaue
Takashimadaira
Nishi-takashimadaira
Edogawabashi
Gokokuji
Korakuen
Kagurazaka
Waseda
Takadanobaba
Ochiai
Nakano-sakaue
Nishi-shinjuku
Tochomae
Shinjuku-sanchome
Shinjuku-gyoemmae
Yoyogi
Kita-sando
Gaiemmae
Aoyama-itchome
Roppongi-itchome
Hiro-o
Naka-meguro
Takanawadai
Sengakuji
Shibakoen
Higashi-ginza
Shintomicho
Ginza-itchome
Takebashi
Jimbocho
Suidobashi
Kudanshita
Hanzomon
Nagatacho
Kojimachi
Ushigome-yanagicho
Ushigome-kagurazaka
Wakamatsu-kawada
Nishi-waseda
Higashi-shinjuku
Akebonobashi
Nerima
Toshimaen
Shin-egota
Ochiai-minami-nagasaki
Higashi-nakano
Nakano-shimbashi
Nakano-fujimicho
Honancho
Shin-nakano
Higashi-koenji
Shin-koenji
Minami-asagaya
Hikawadai
Heiwadai
Chikatetsu-narimasu
Chikatetsu-akatsuka
Kotake-mukaihara
Senkawa
Kanamecho
Shin-otsuka
Myogadani
Higashi-ikebukuro
Zoshigaya
Tsukishima
Shin-kiba
Tatsumi
Toyosu
Kiba
Kasai
Nishi-kasai
Urayasu
Minami-gyotoku
Gyotoku
Myoden
Baraki-nakayama
Nishi-funabashi
Tokyo Metro

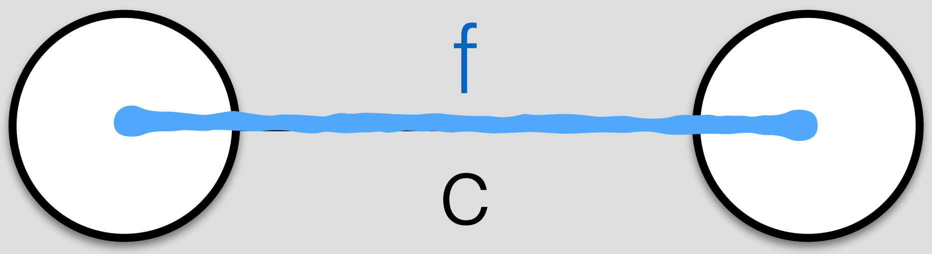

s
t
1
3
3
1
1
2
1
1



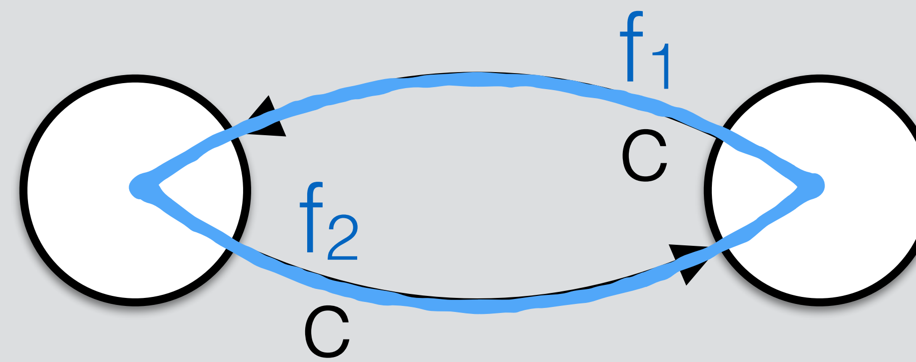
1
3
1
2
1
1
2
1
1
2
1
1
3
1
1

Flows in undirected graphs

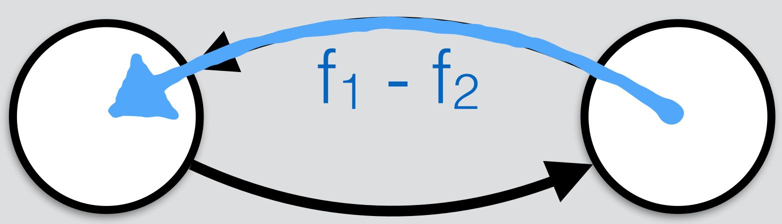
Which direction is flow?



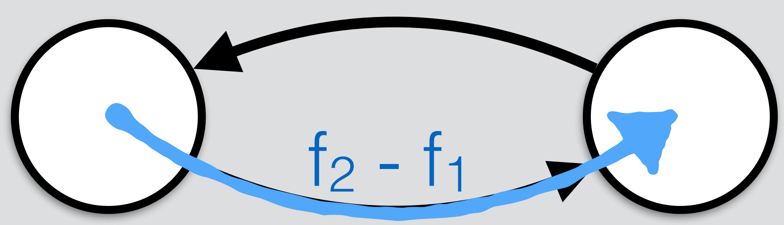
Directed equivalent:



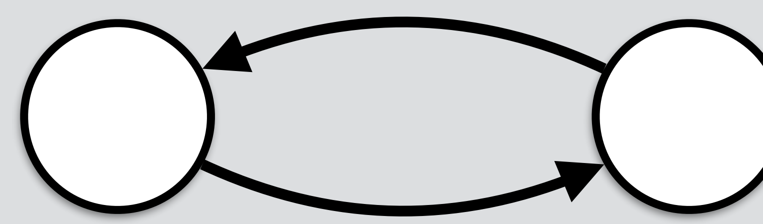
Three cases of equivalent flows:



if $f_1 > f_2$

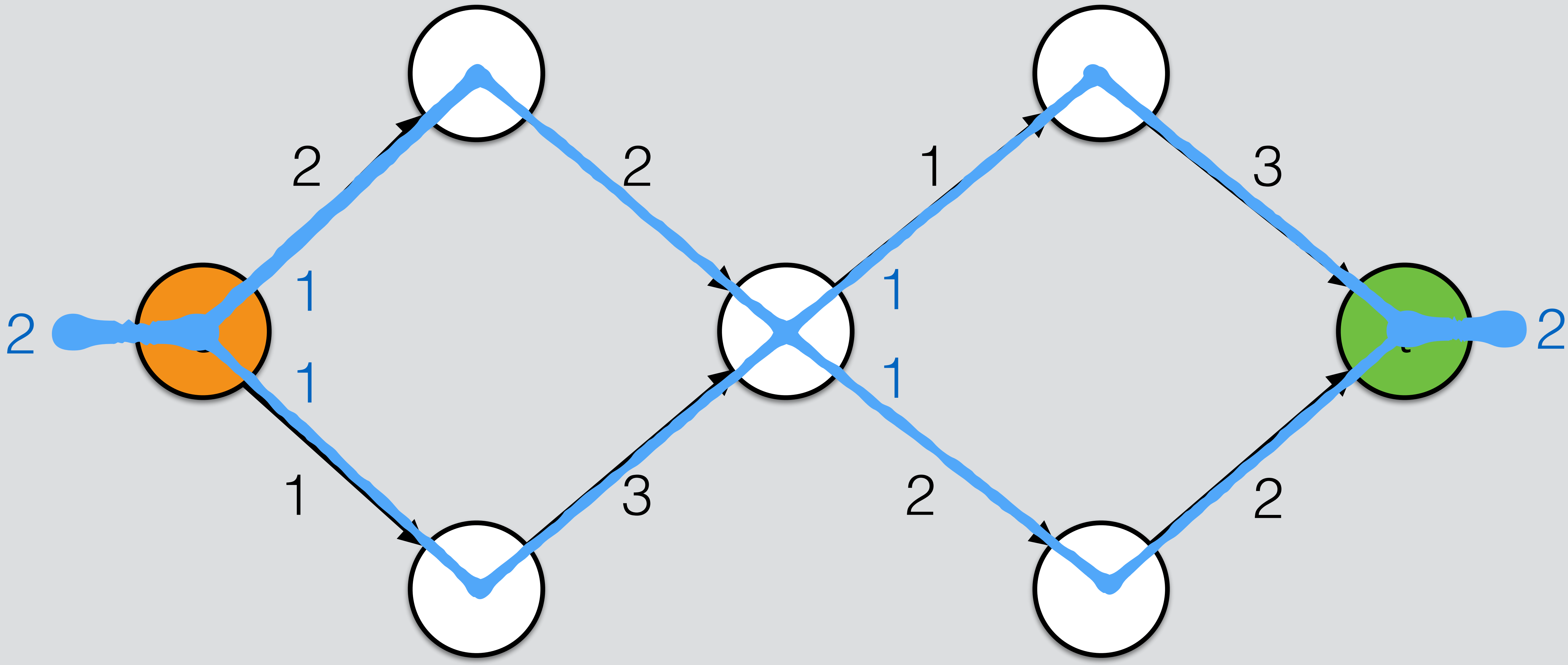


if $f_1 < f_2$

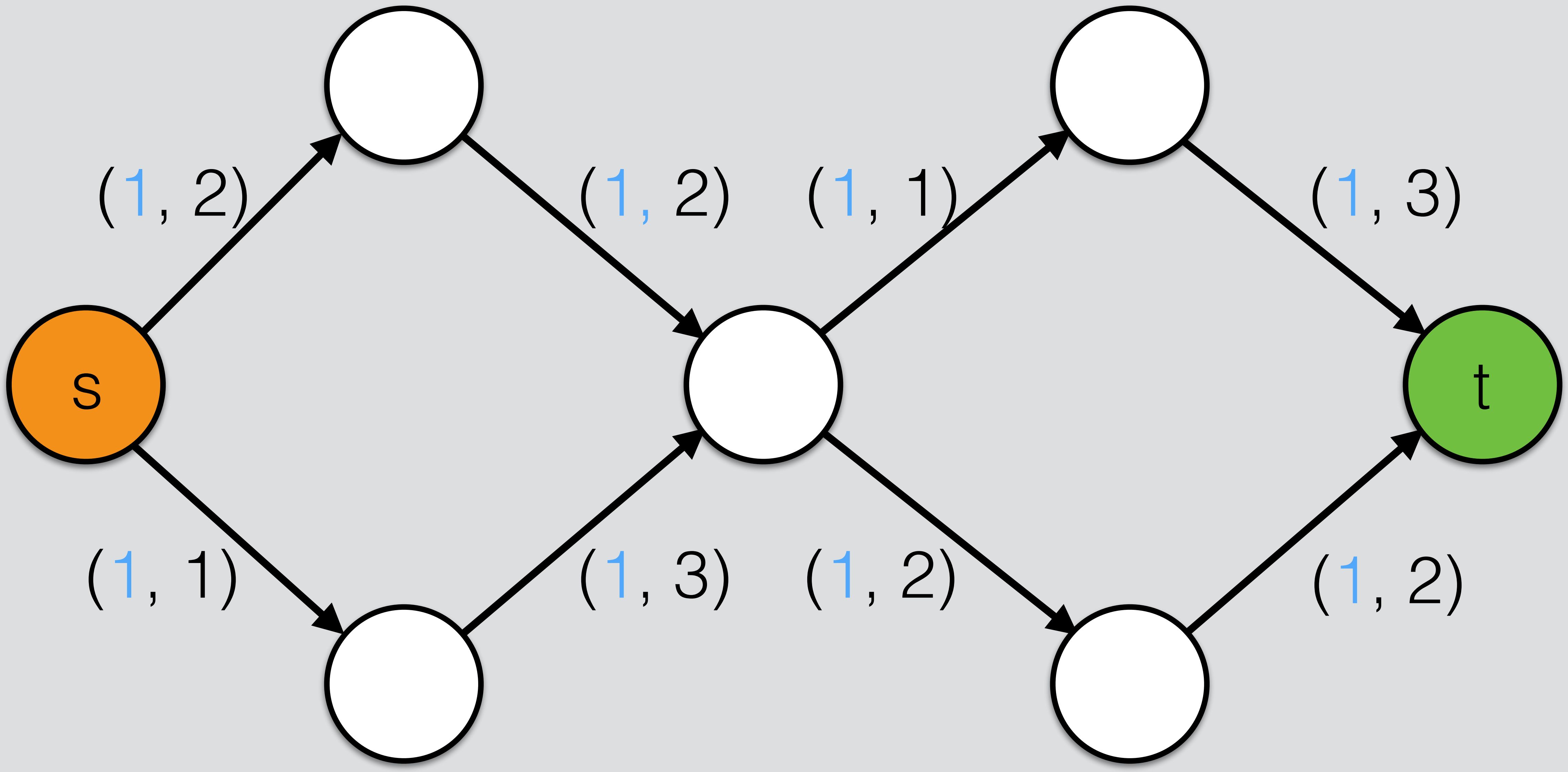


if $f_1 = f_2$

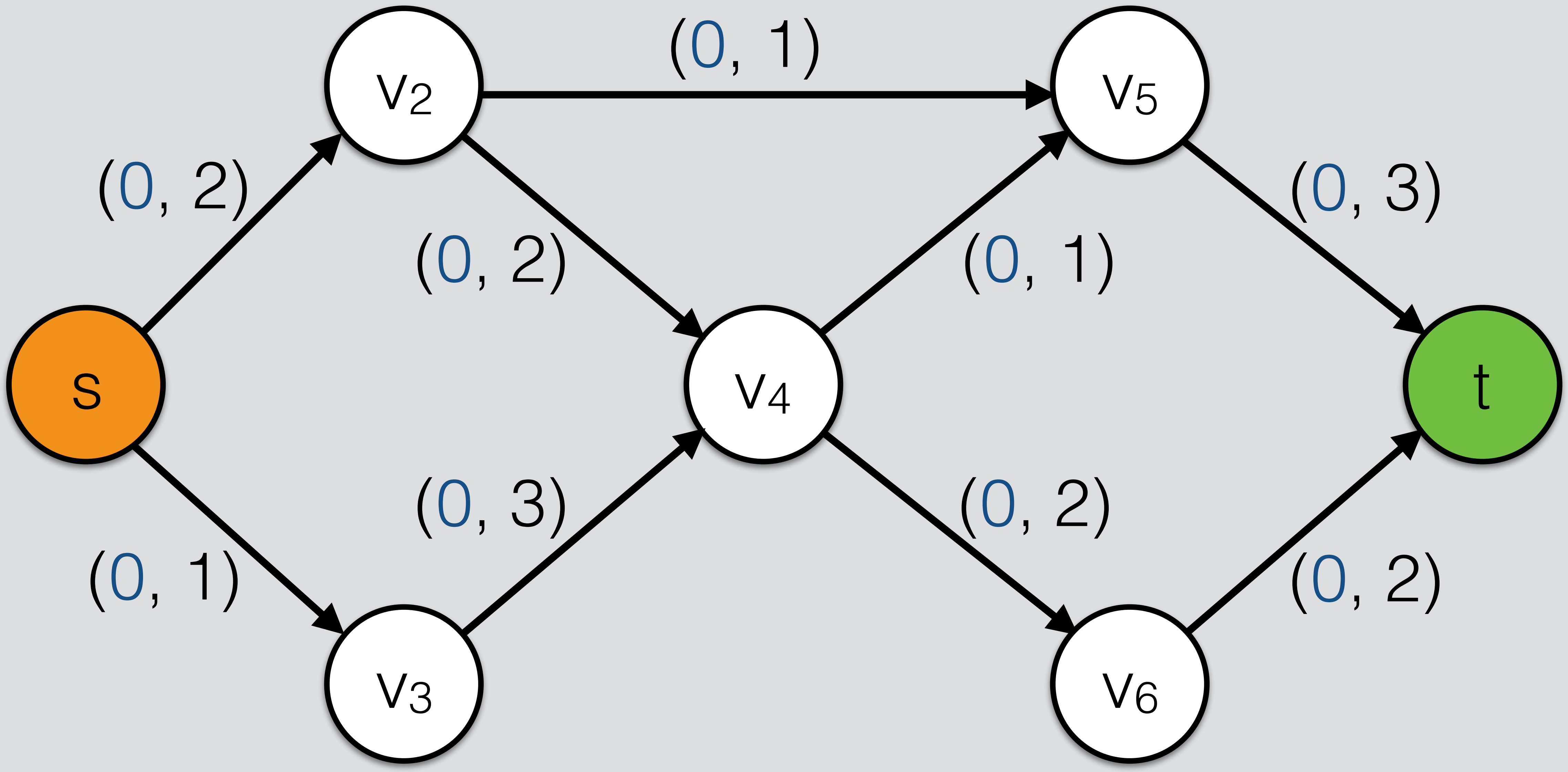
Flow Notation



2
2
2
1
3
1
3
2
2
1
1
1
1
2

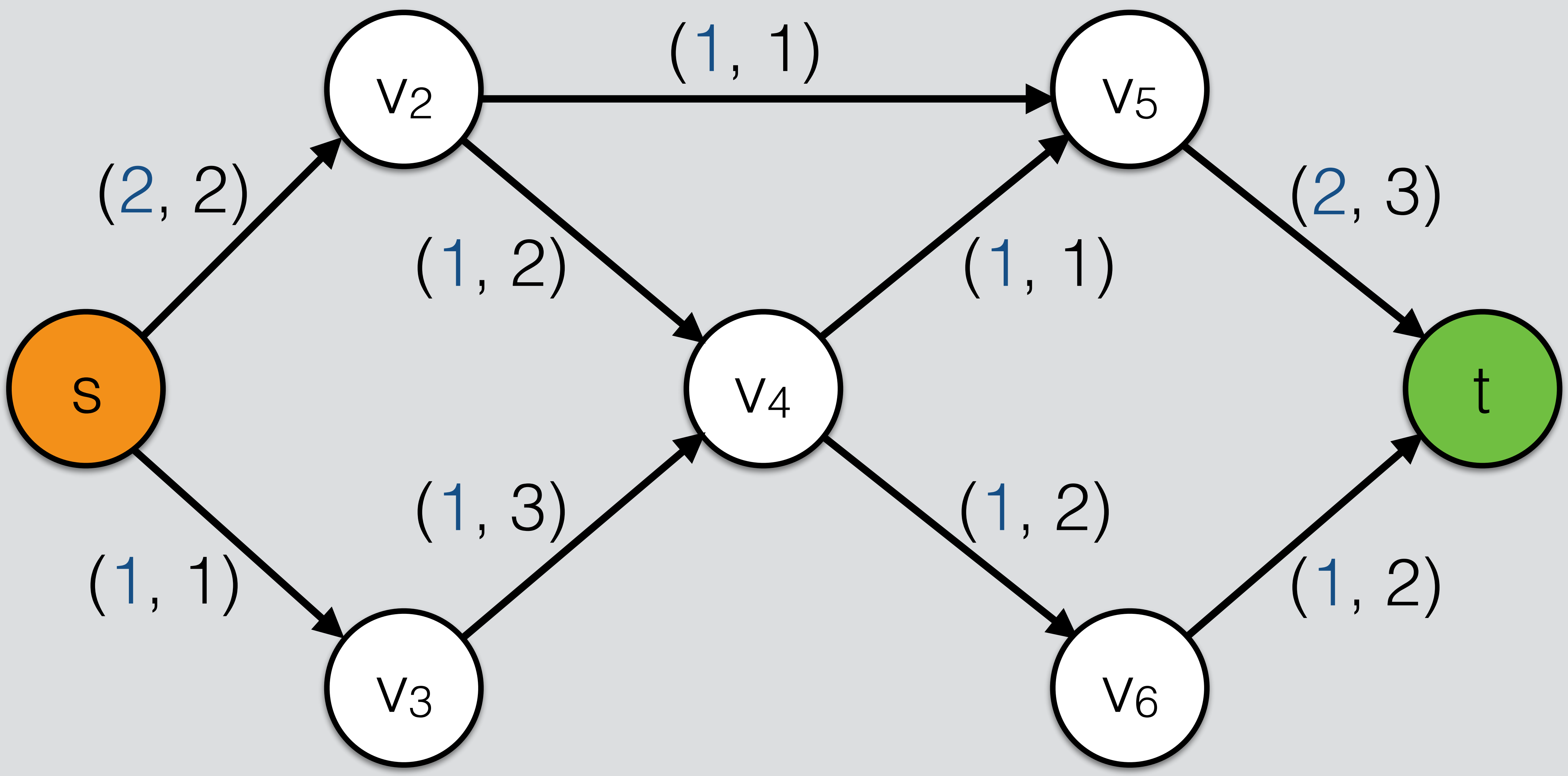

(1, 2)
(1, 2)
(1, 1)
(1, 3)
s
t
(1, 1)
(1, 3)
(1, 2)
(1, 2)

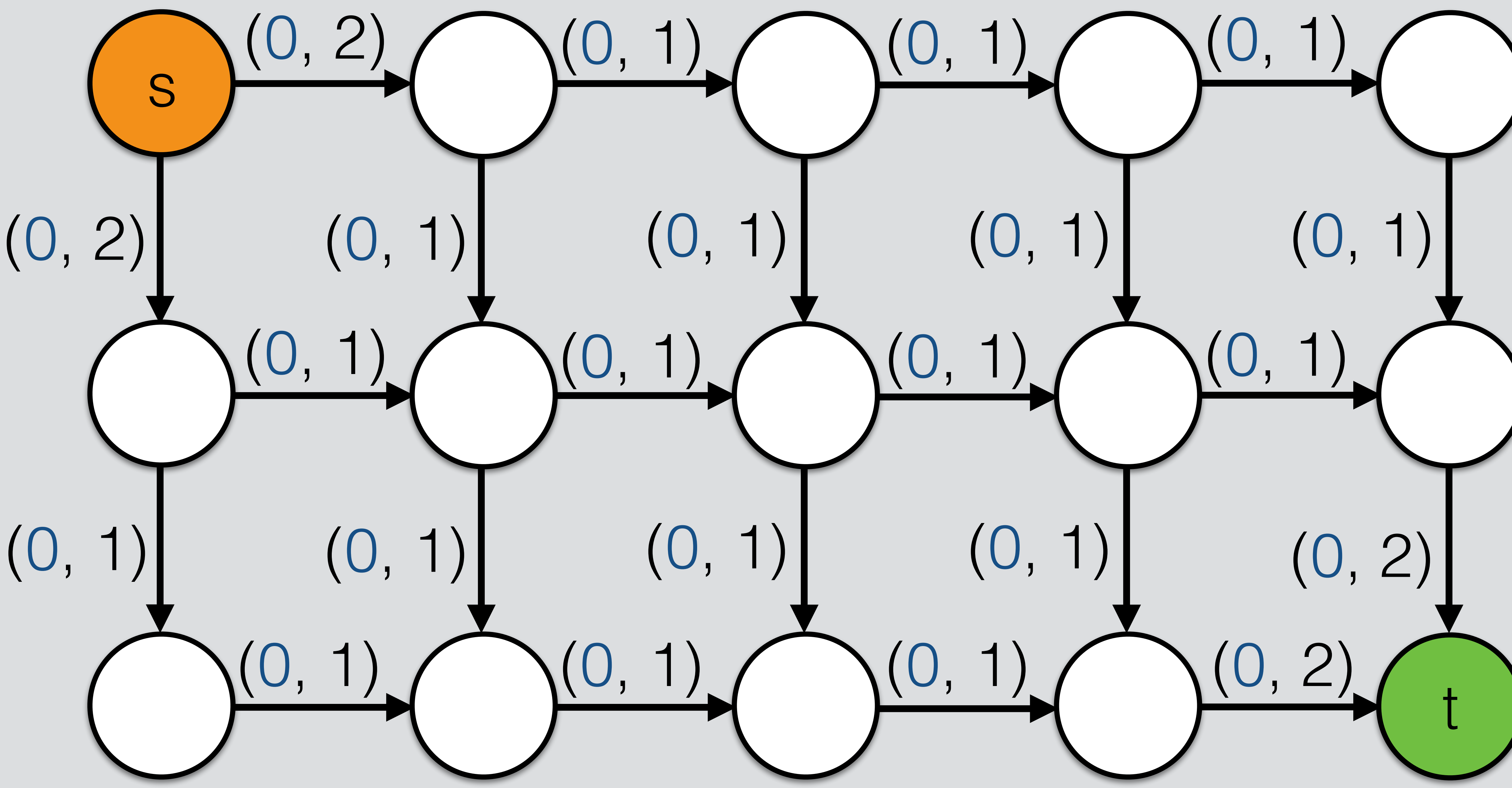
Maximum Flows



s
v2
v3
v4
v5
v6
t
(0, 2)
(0, 1)
(0, 1)
(0, 2)
(0, 3)
(0, 1)
(0, 2)
(0, 3)
(0, 2)

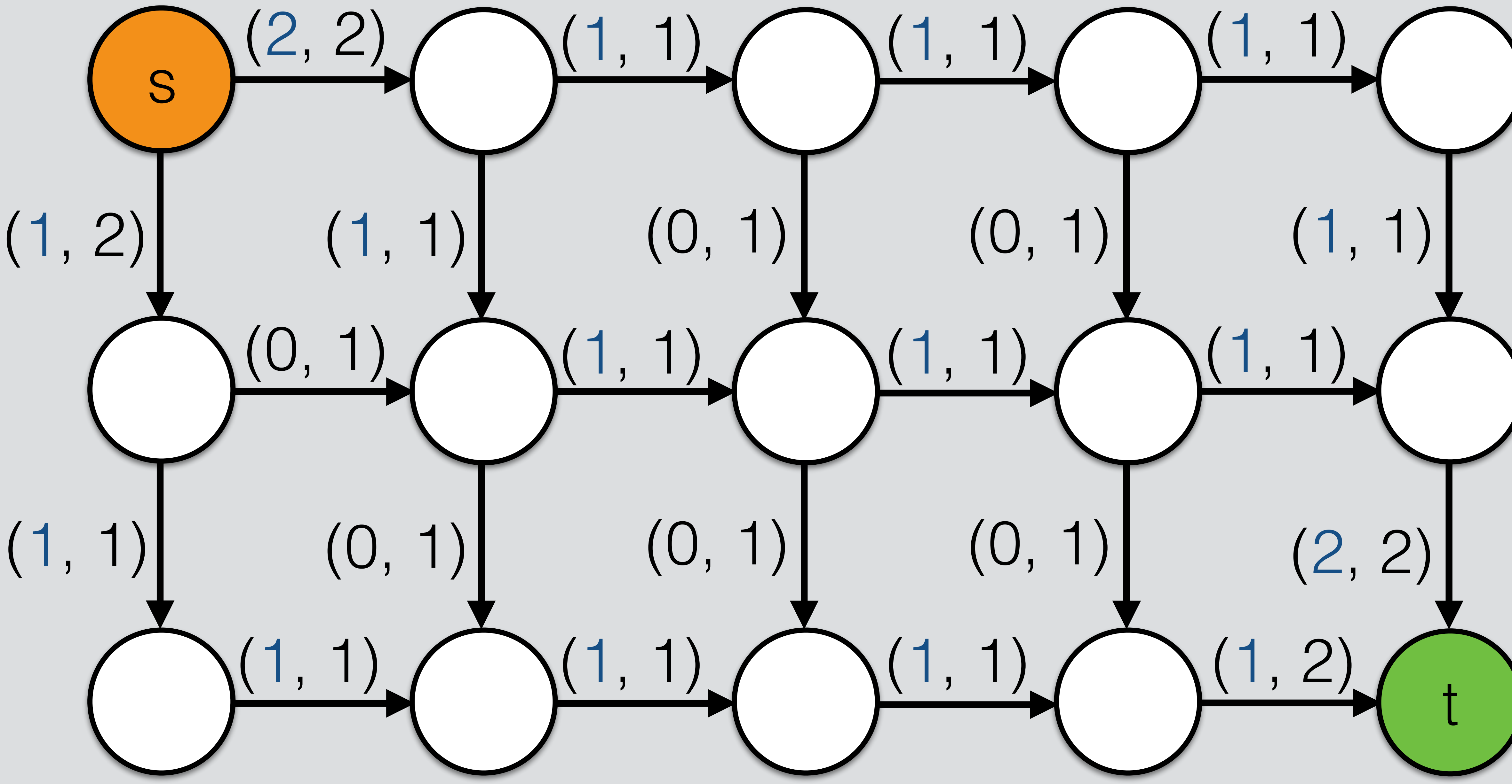
Maximum flow of 3



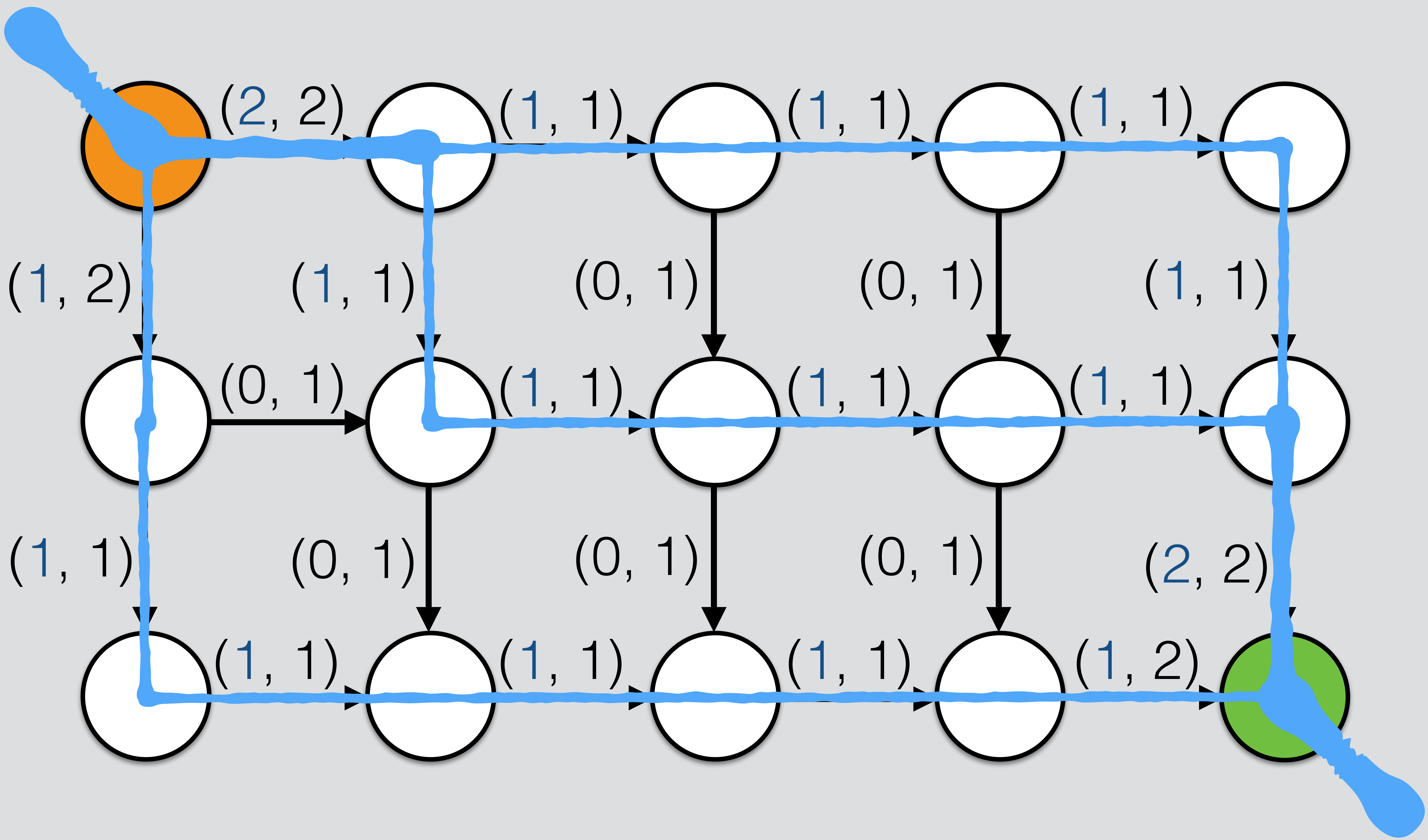


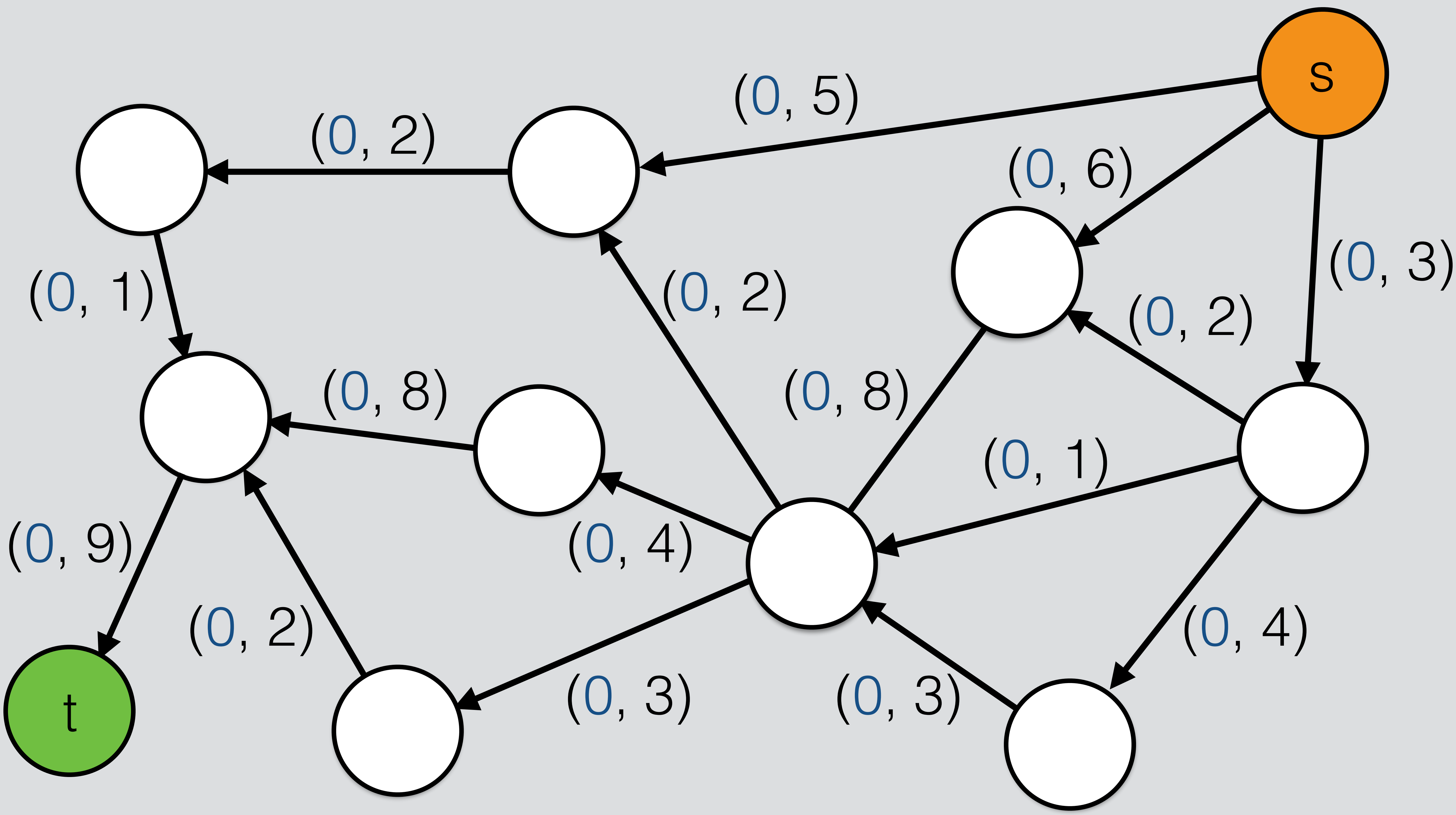
s
(0, 2)
(0, 1)
(0, 1)
(0, 1)
(0, 2)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 1)
(0, 2)
(0, 1)
(0, 1)
(0, 1)
(0, 2)
t

Maximum flow of 3

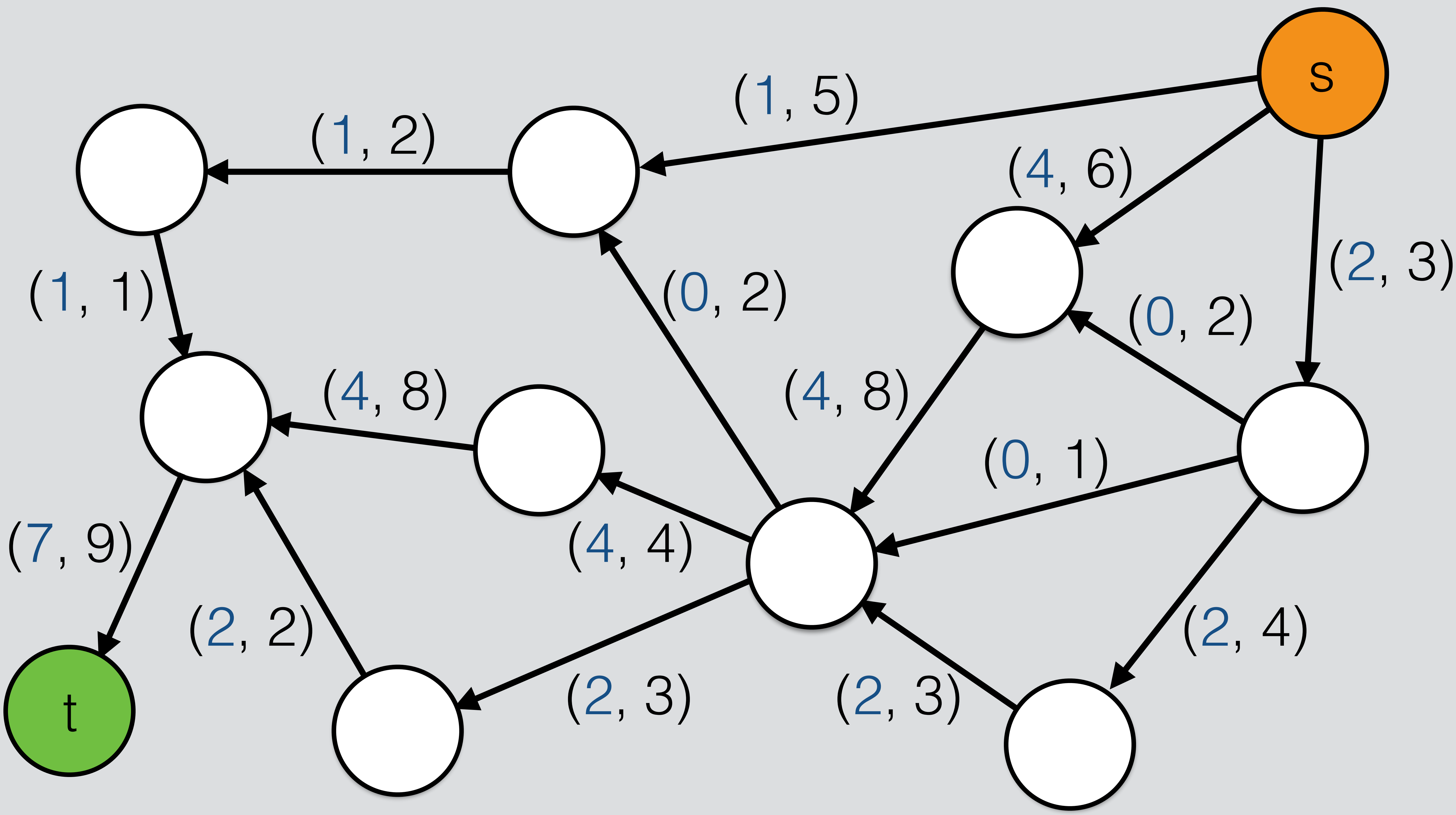


Maximum flow of 3

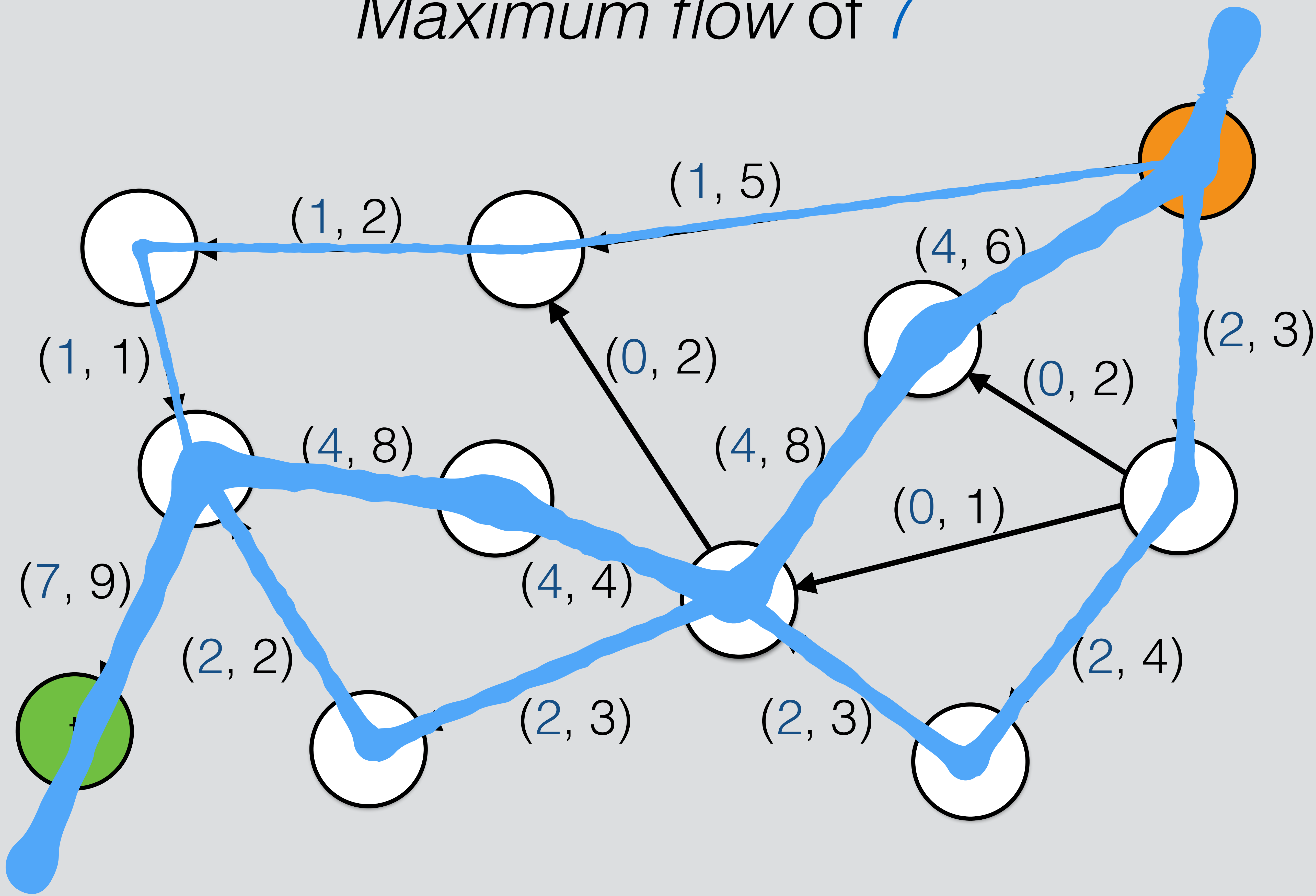



s
t
(0, 5)
(0, 2)
(0, 6)
(0, 3)
(0, 1)
(0, 2)
(0, 2)
(0, 8)
(0, 8)
(0, 1)
(0, 9)
(0, 4)
(0, 2)
(0, 4)
(0, 3)
(0, 3)

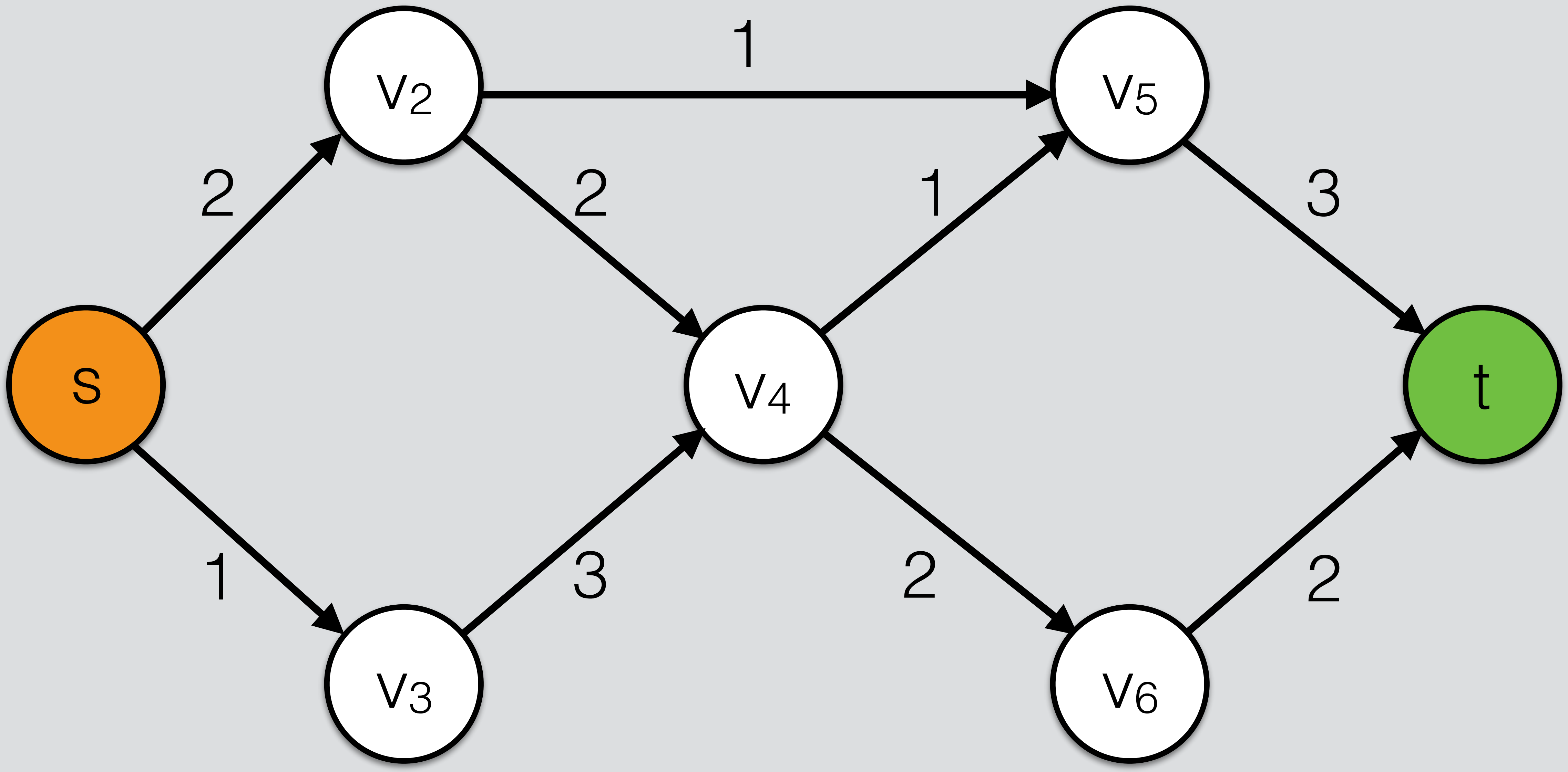
Maximum flow of 7



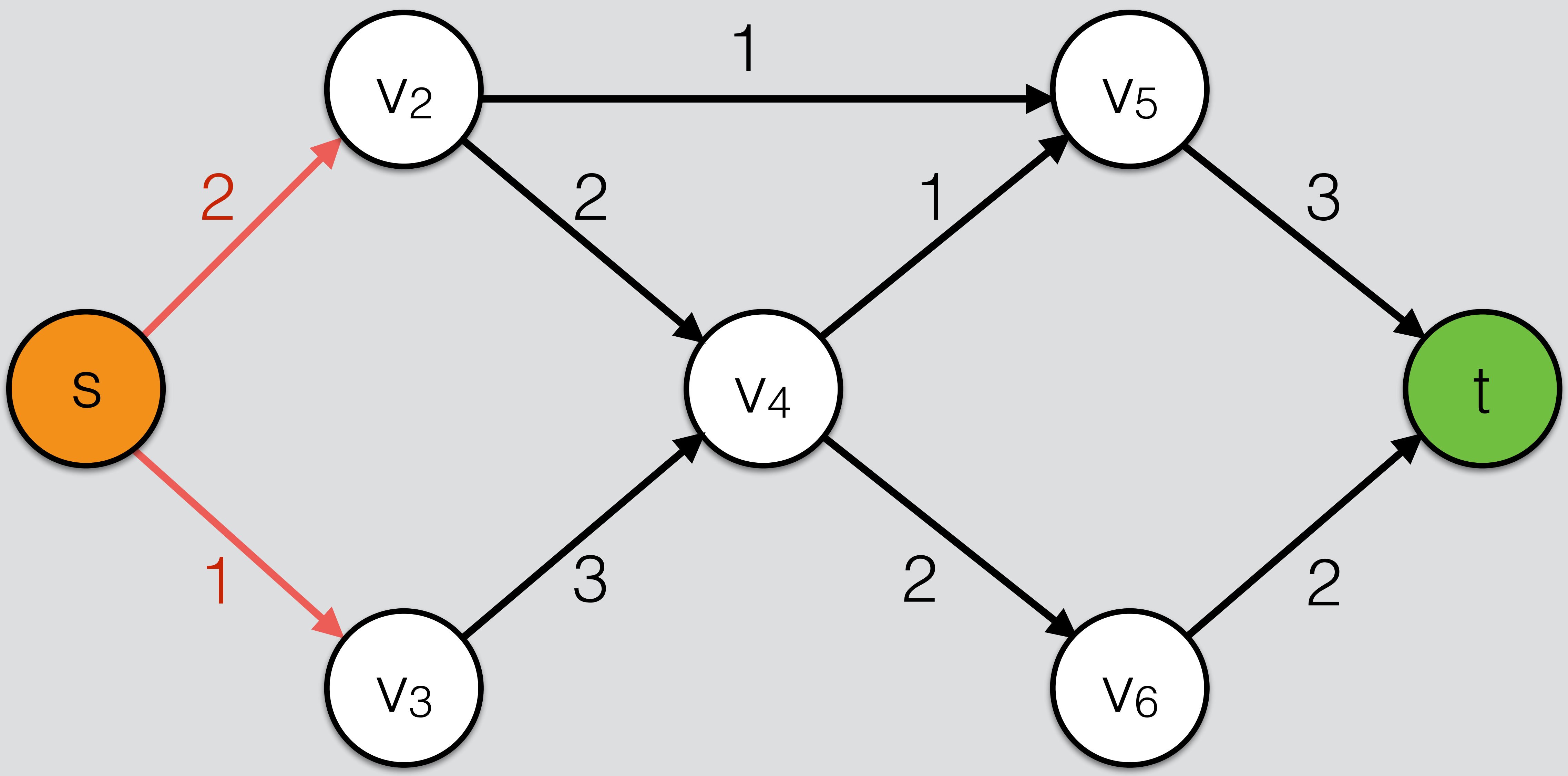
Maximum flow of 7

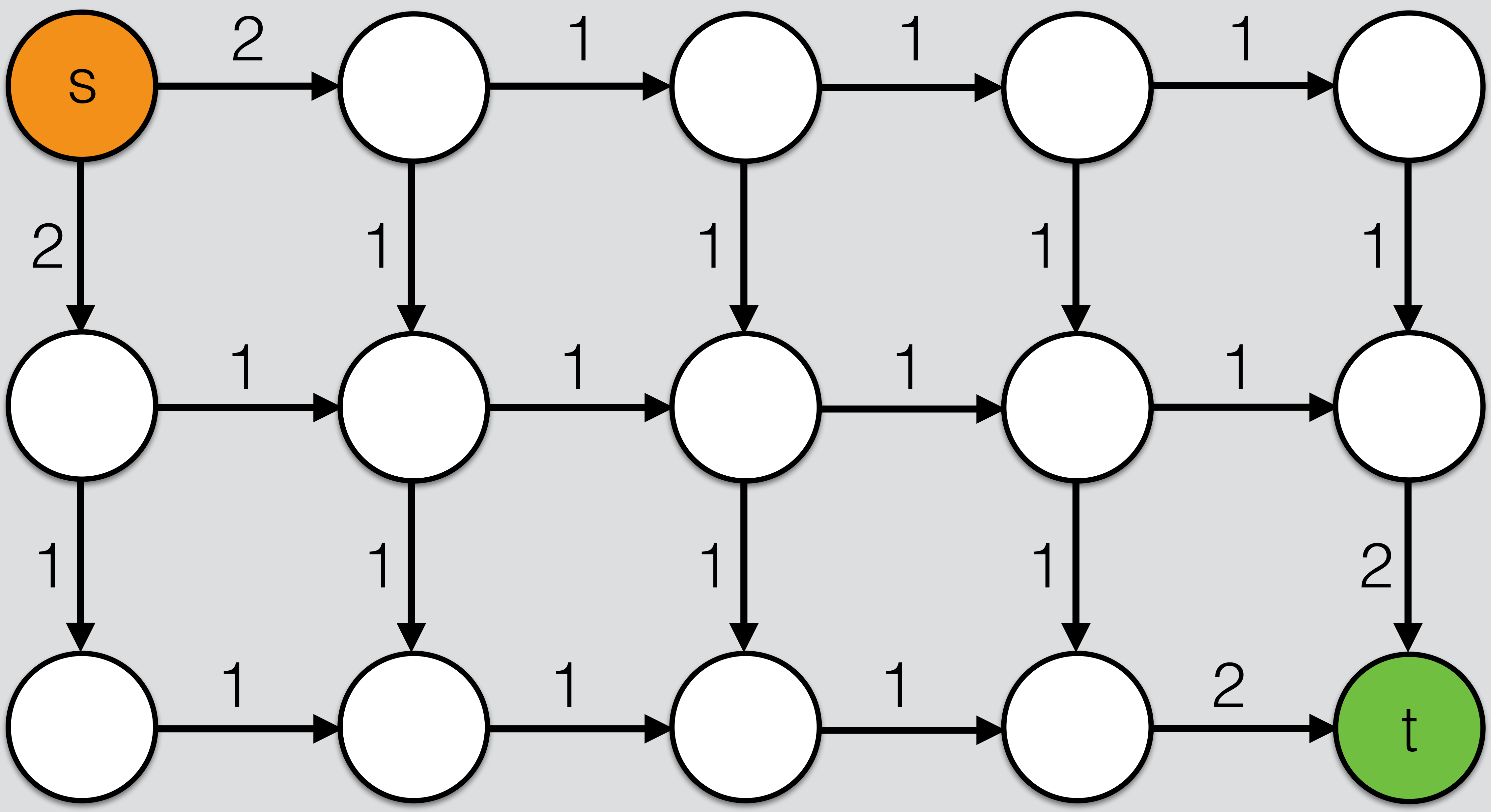


Minimum Cuts

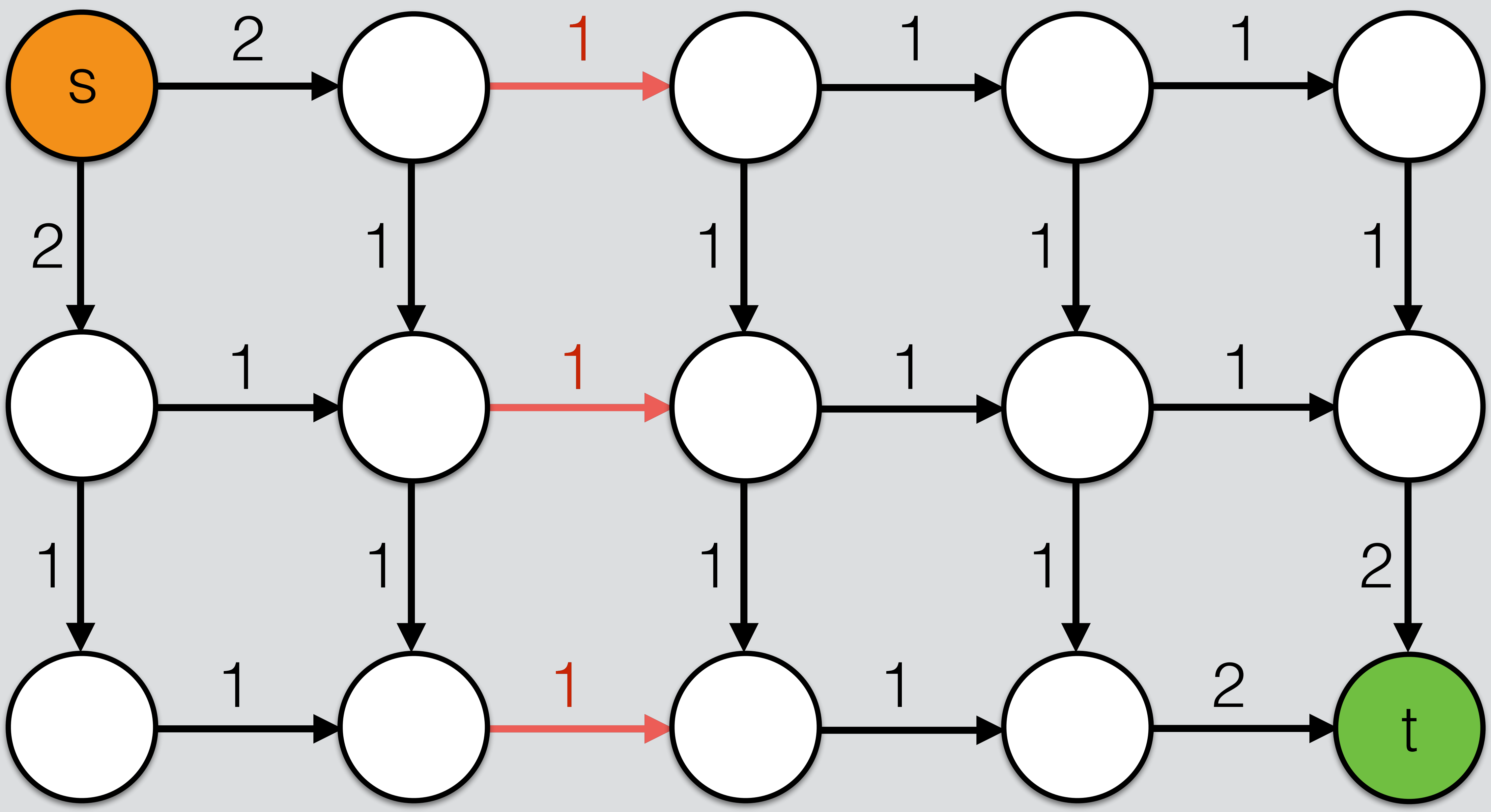

1
2
2
1
3
s
v2
v5
v4
t
1
3
2
2
v3
v6

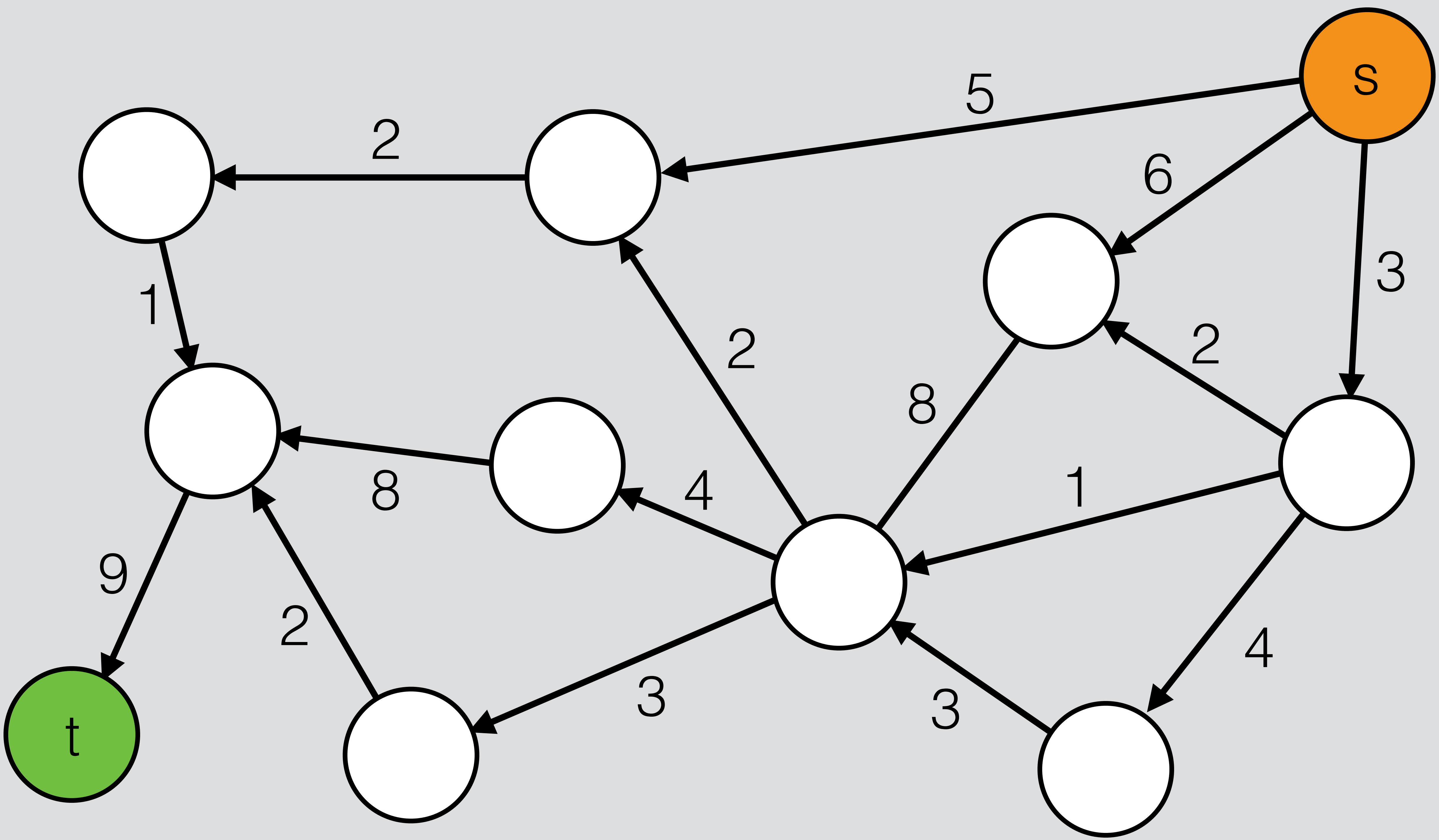
Minimum edge cut of 3



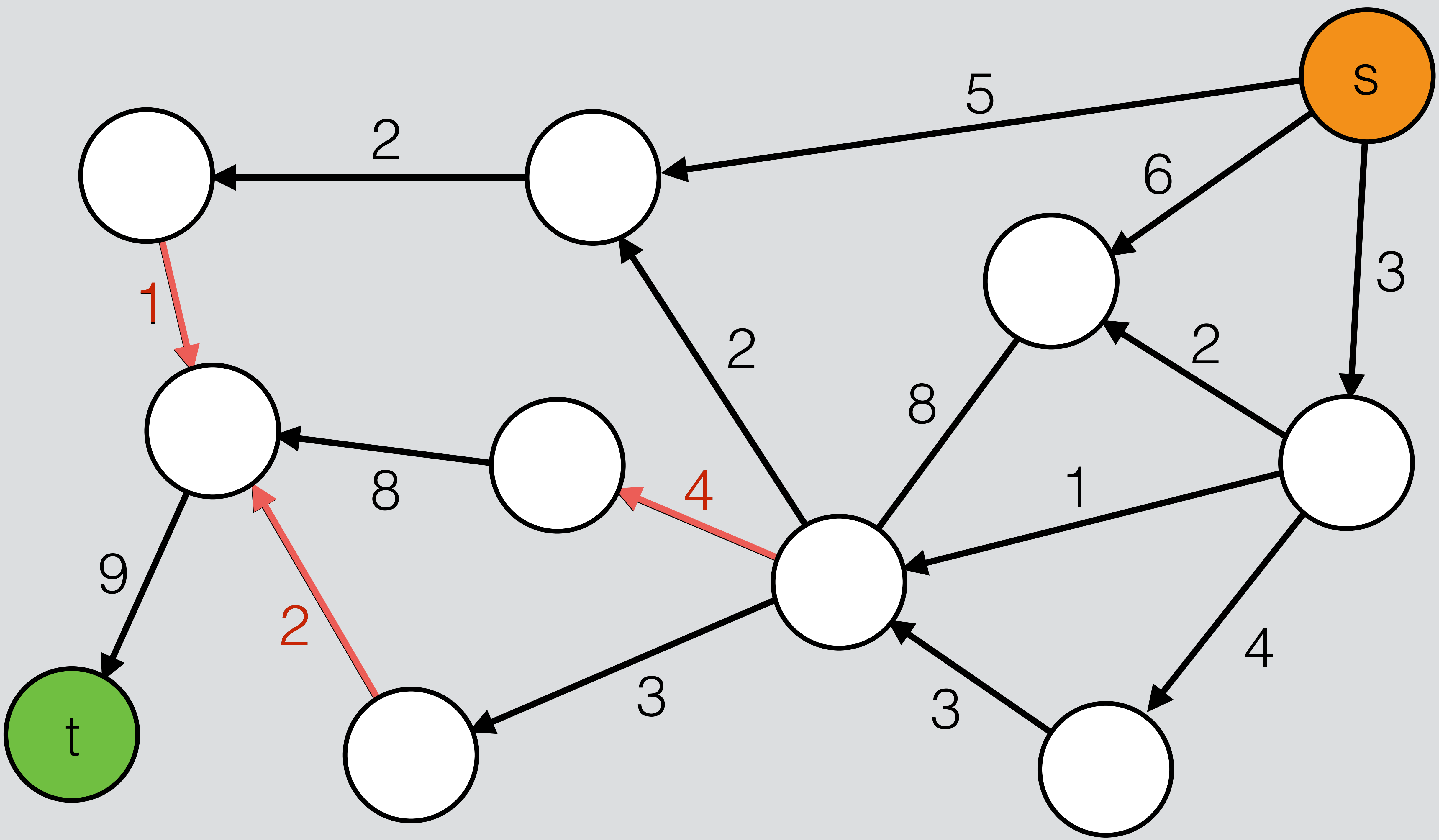

s
2
1
1
1
2
1
1
1
1
1
1
1
1
1
1
1
1
2
1
1
1
2
t

Minimum edge cut of 3

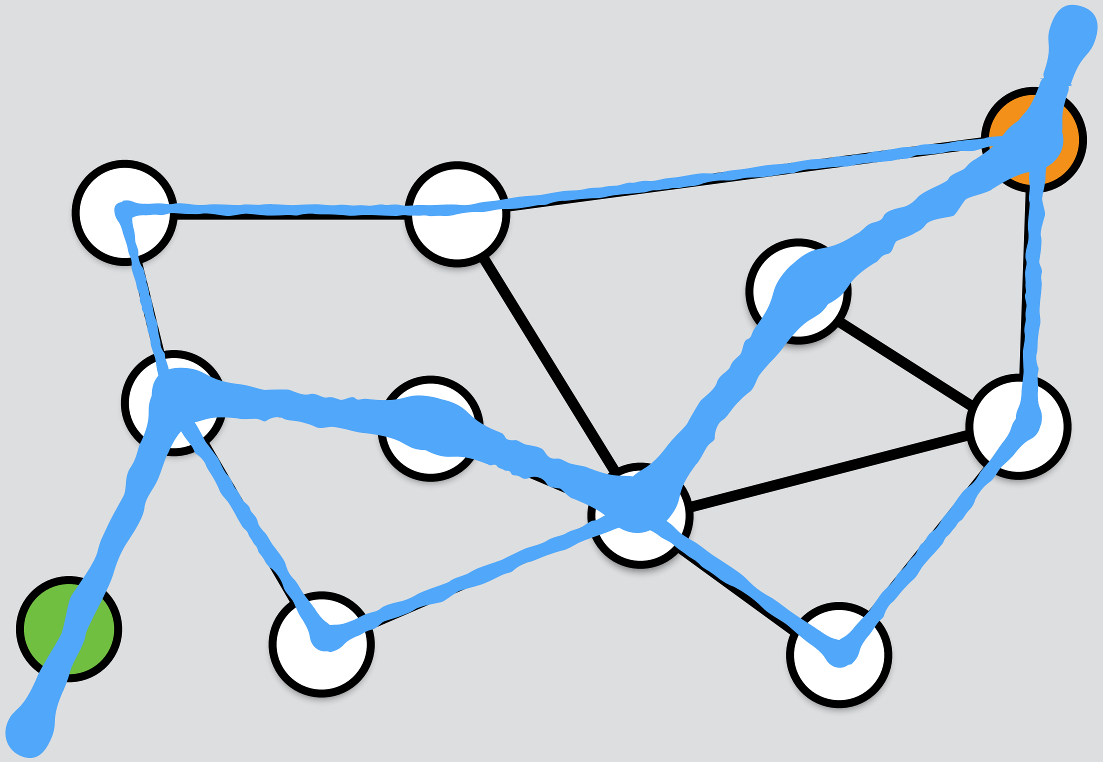



s
t
5
6
3
2
2
1
2
8
2
8
4
1
9
2
3
3
4

Minimum edge cut of 7

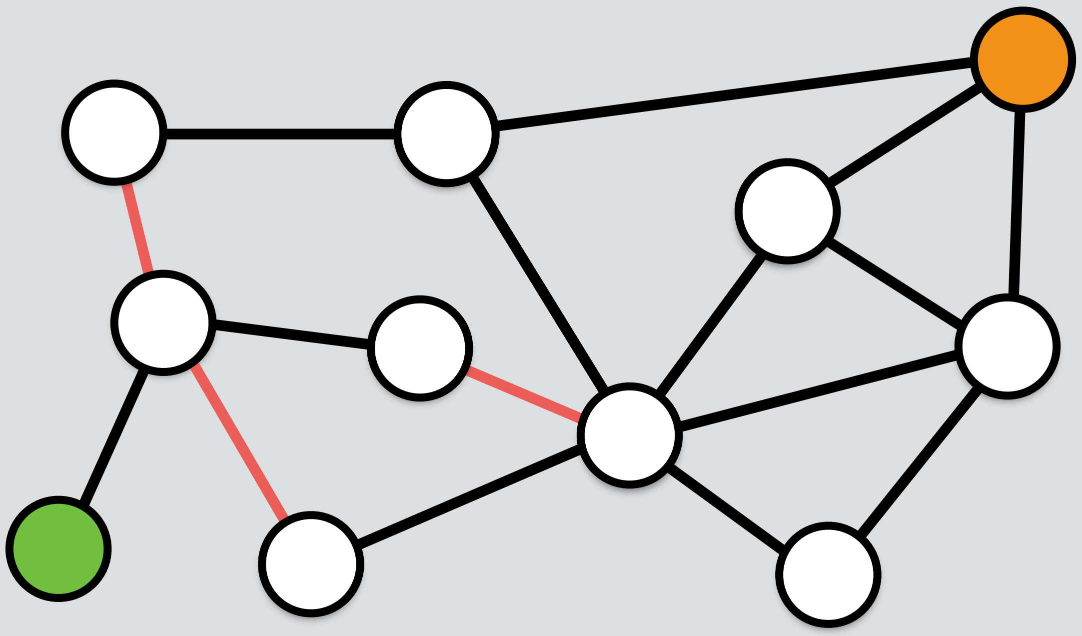


For any graph with vertices s and t



maximum flow
from s to t

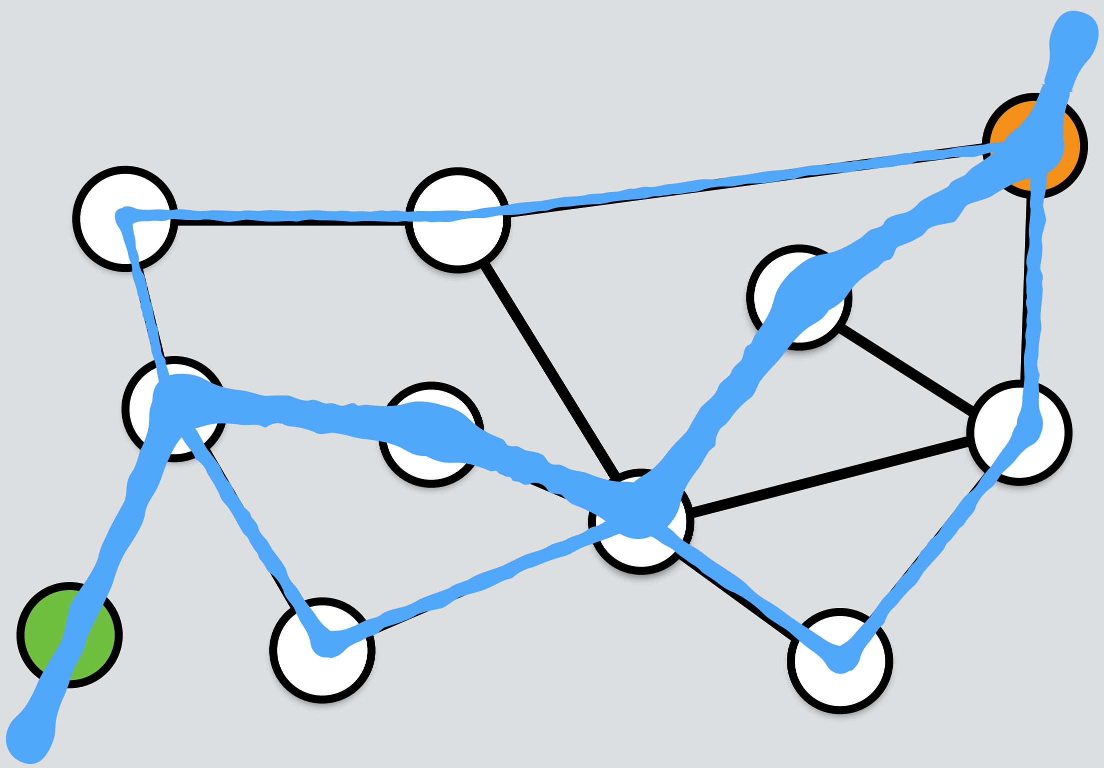
=



minimum edge cut
separating s from t

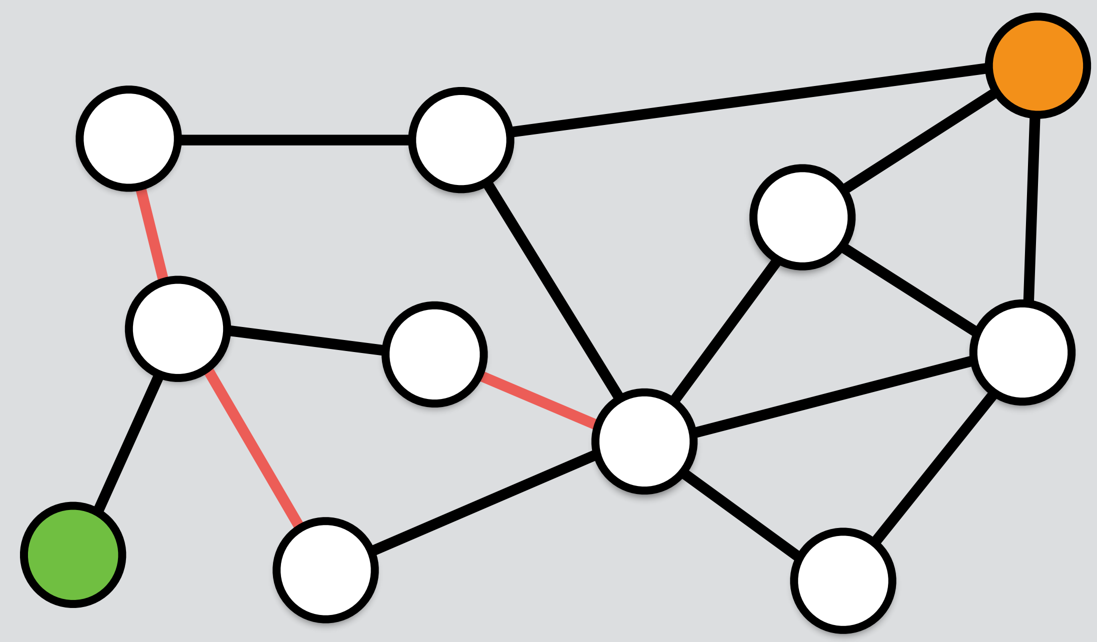
Max-flow min-cut theorem

For any graph with vertices s and t



maximum flow
from s to t

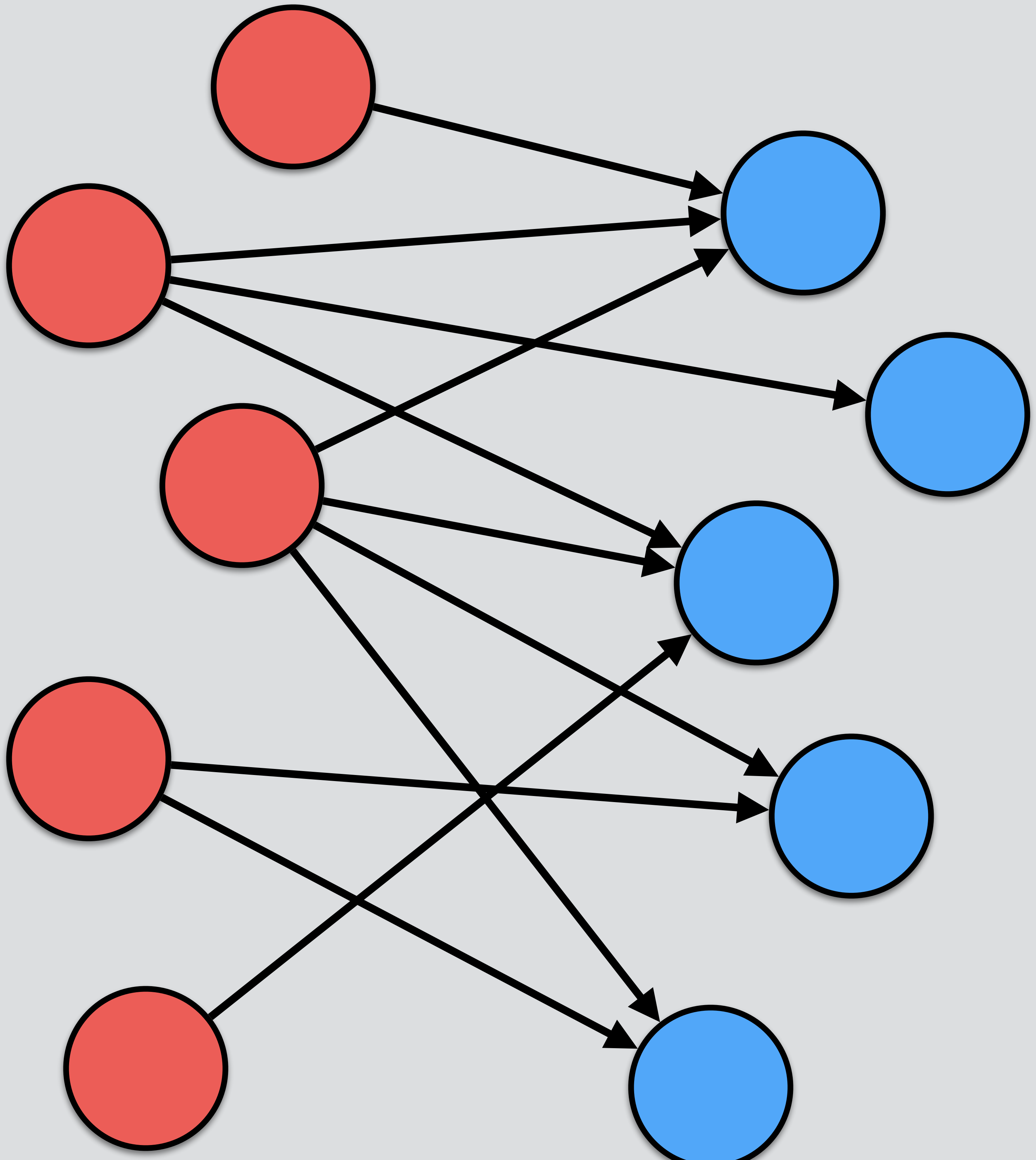
=



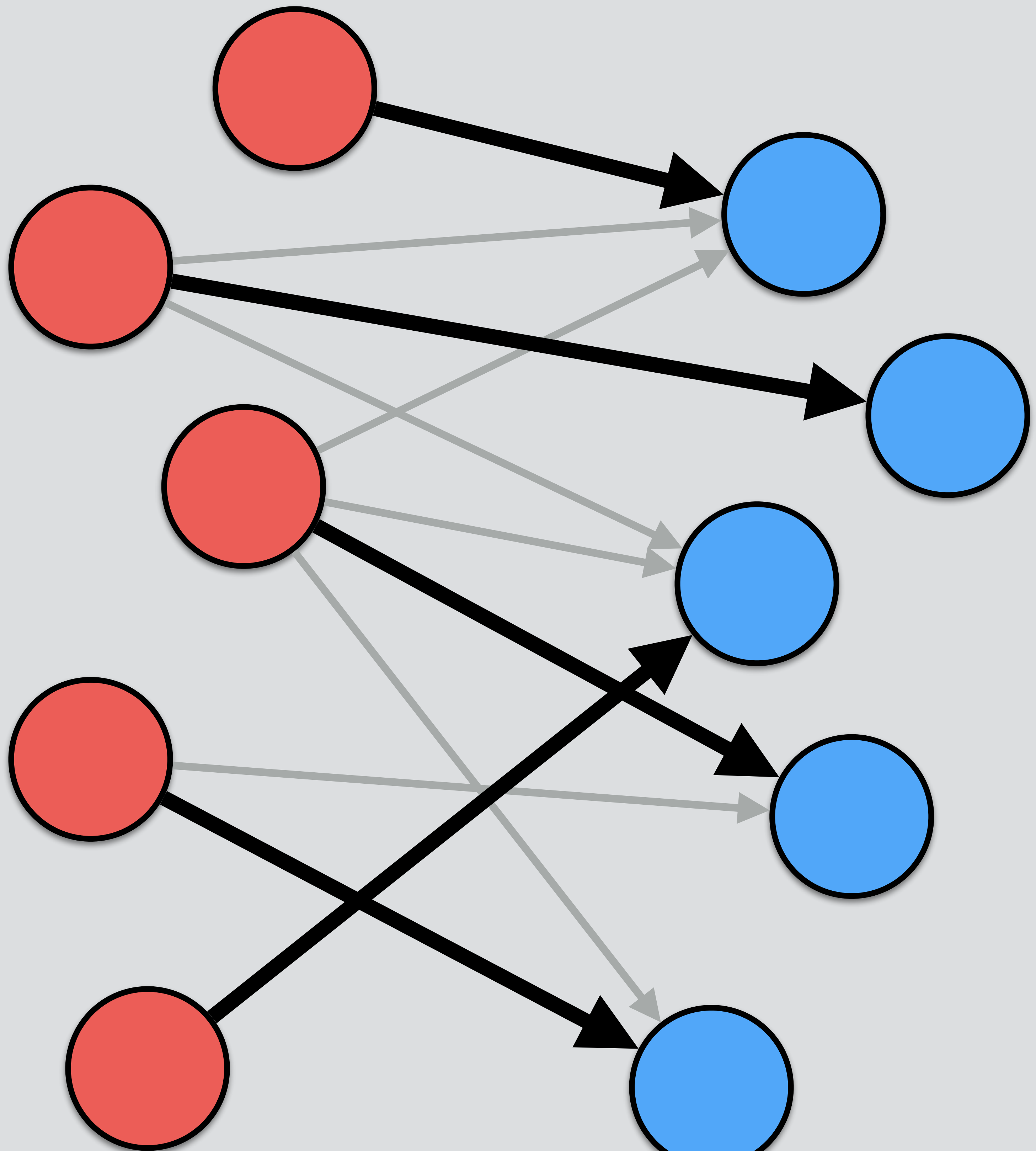
minimum edge cut
separating s from t

Bipartite
Perfect Matching

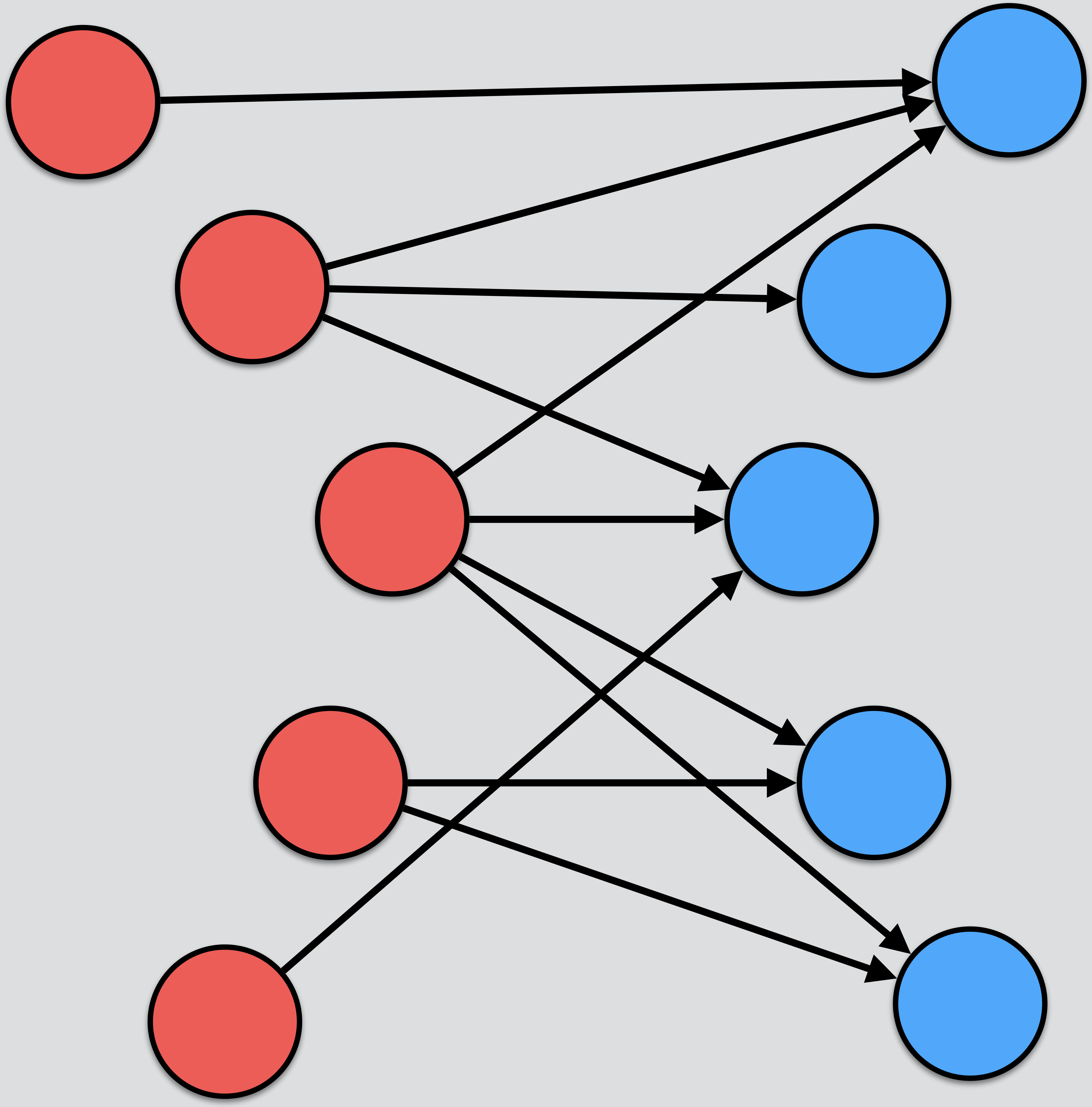
Bipartite graph

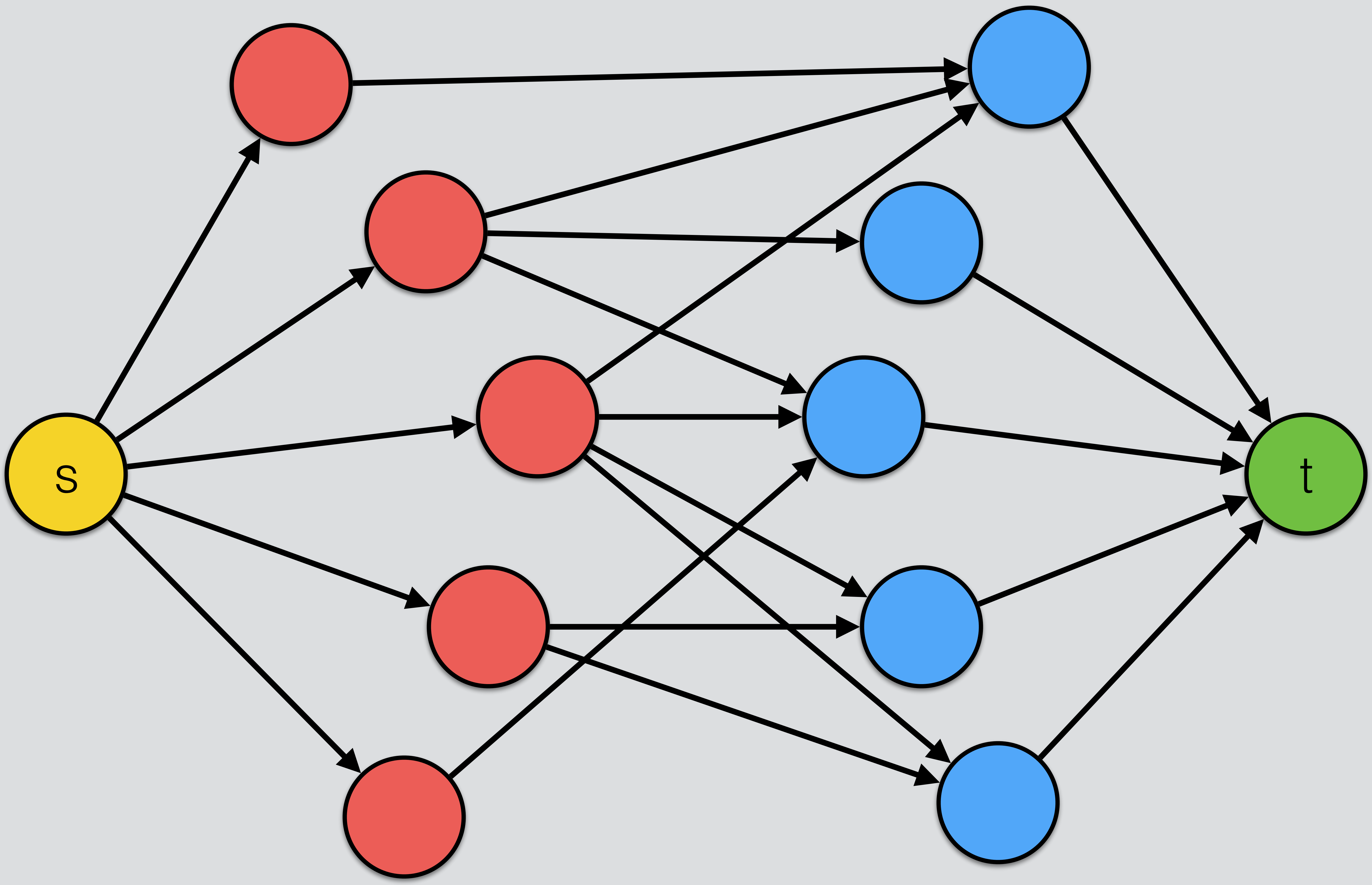


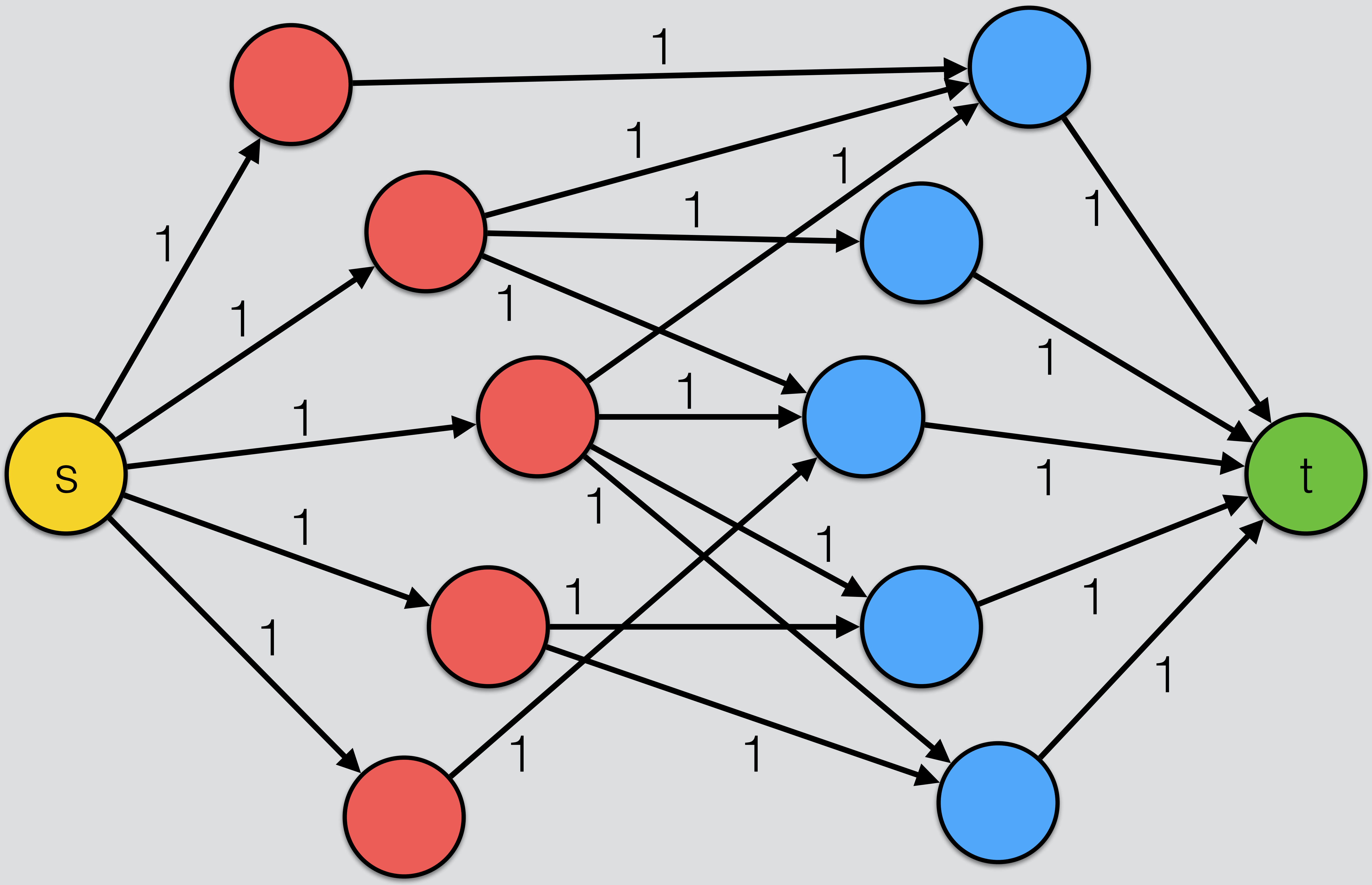
Perfect Matching

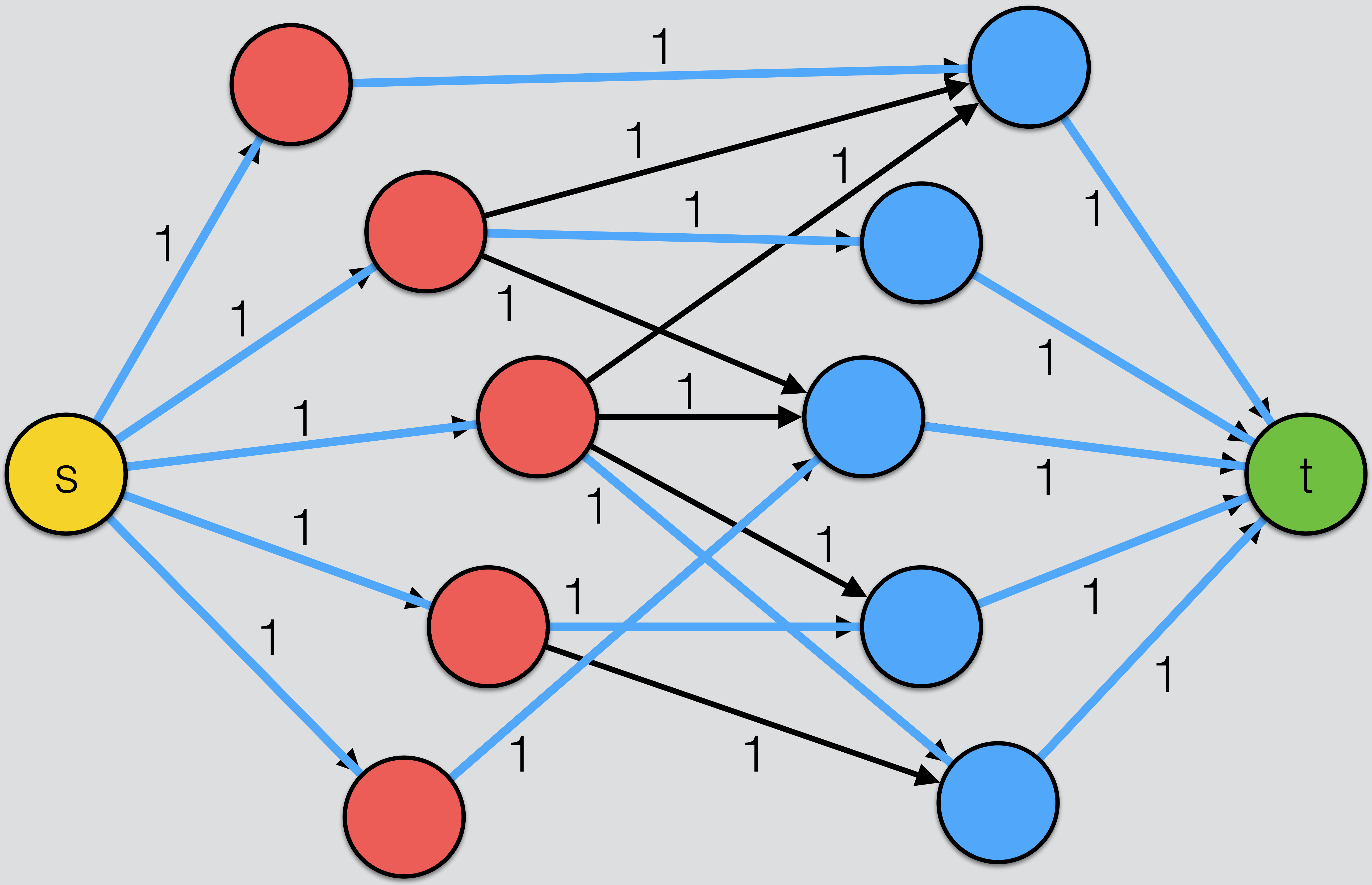


Perfect Matching




s
t

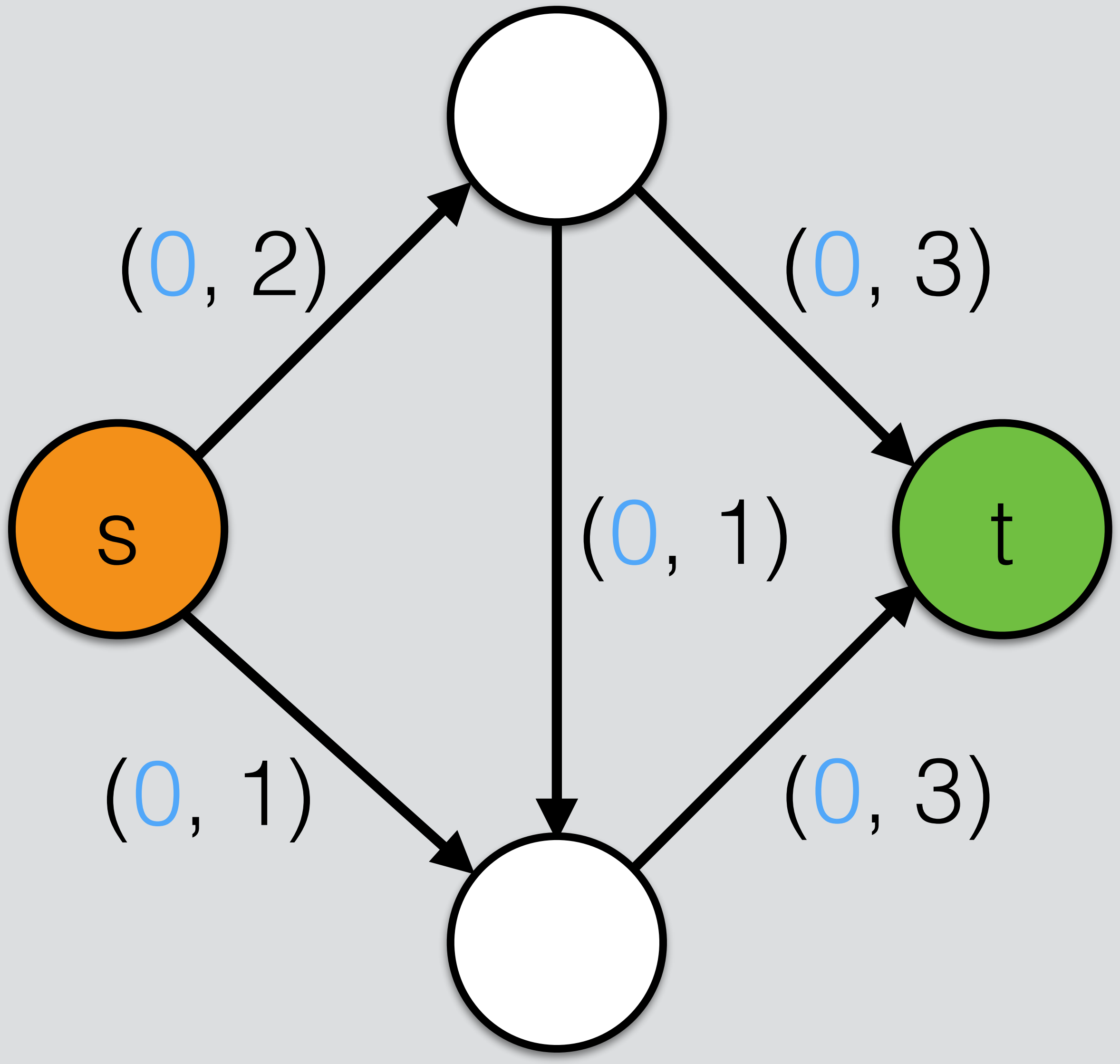

s
t
1



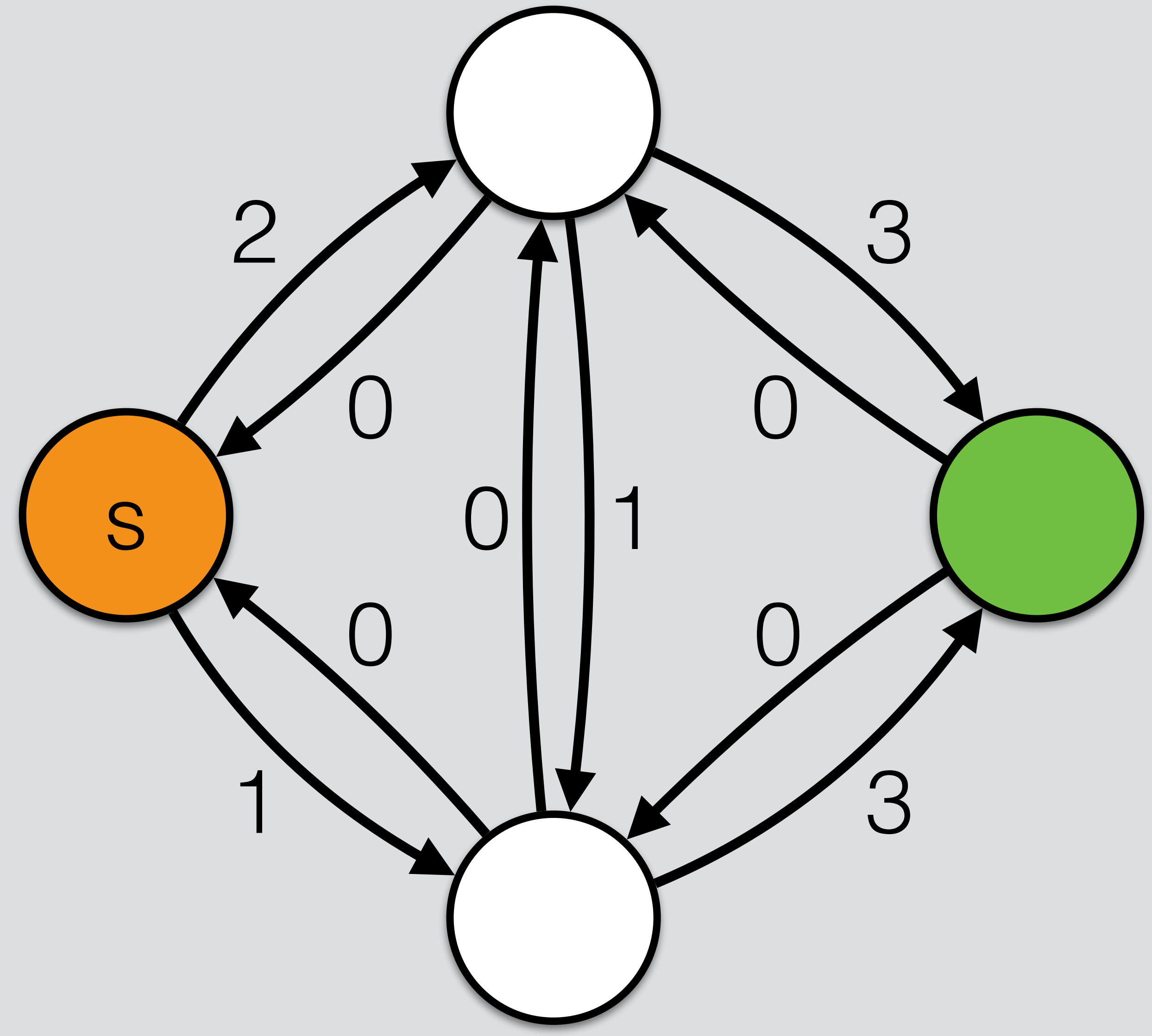
s
t
1

Residual Graphs

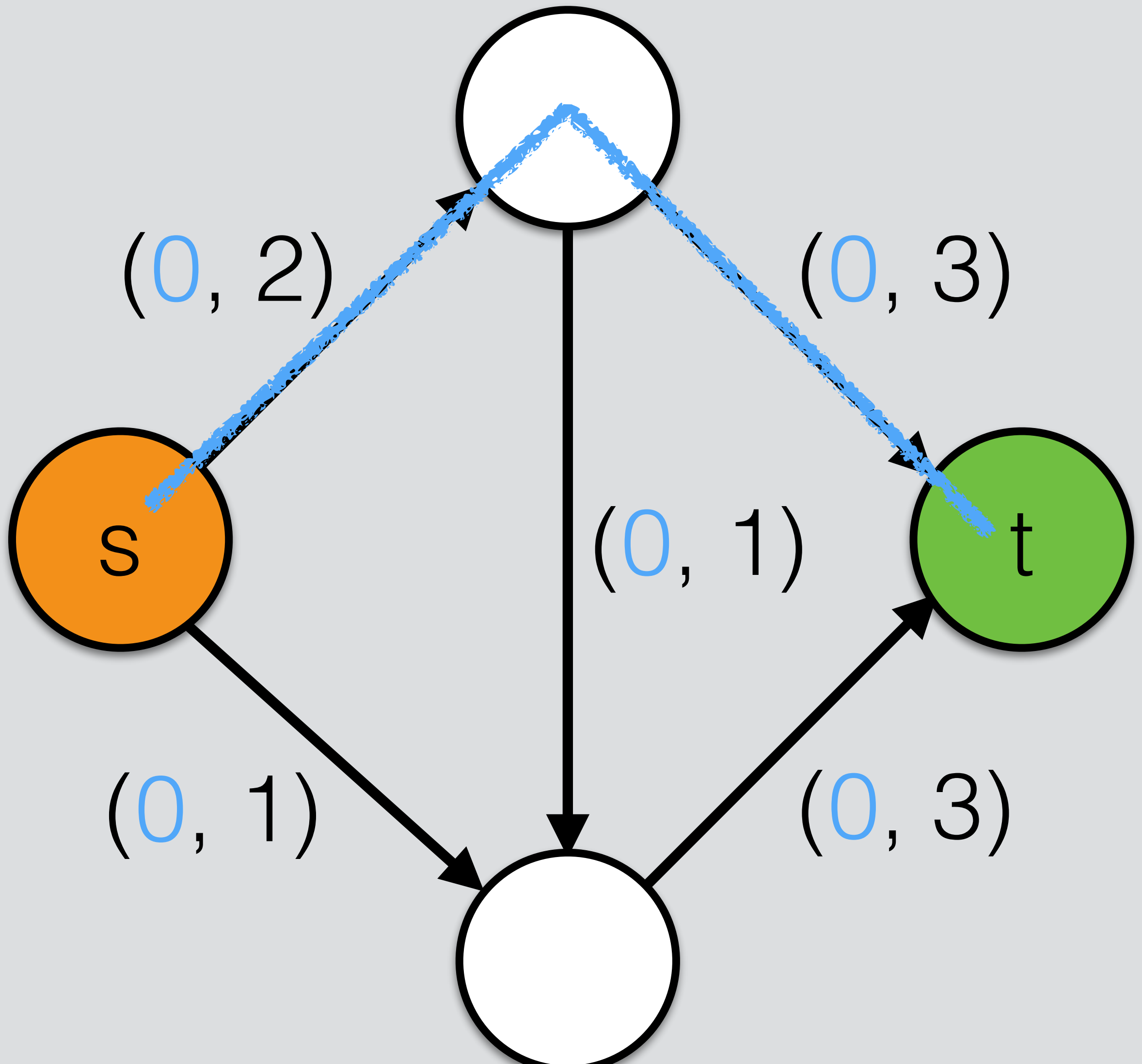
Graph (with flow)



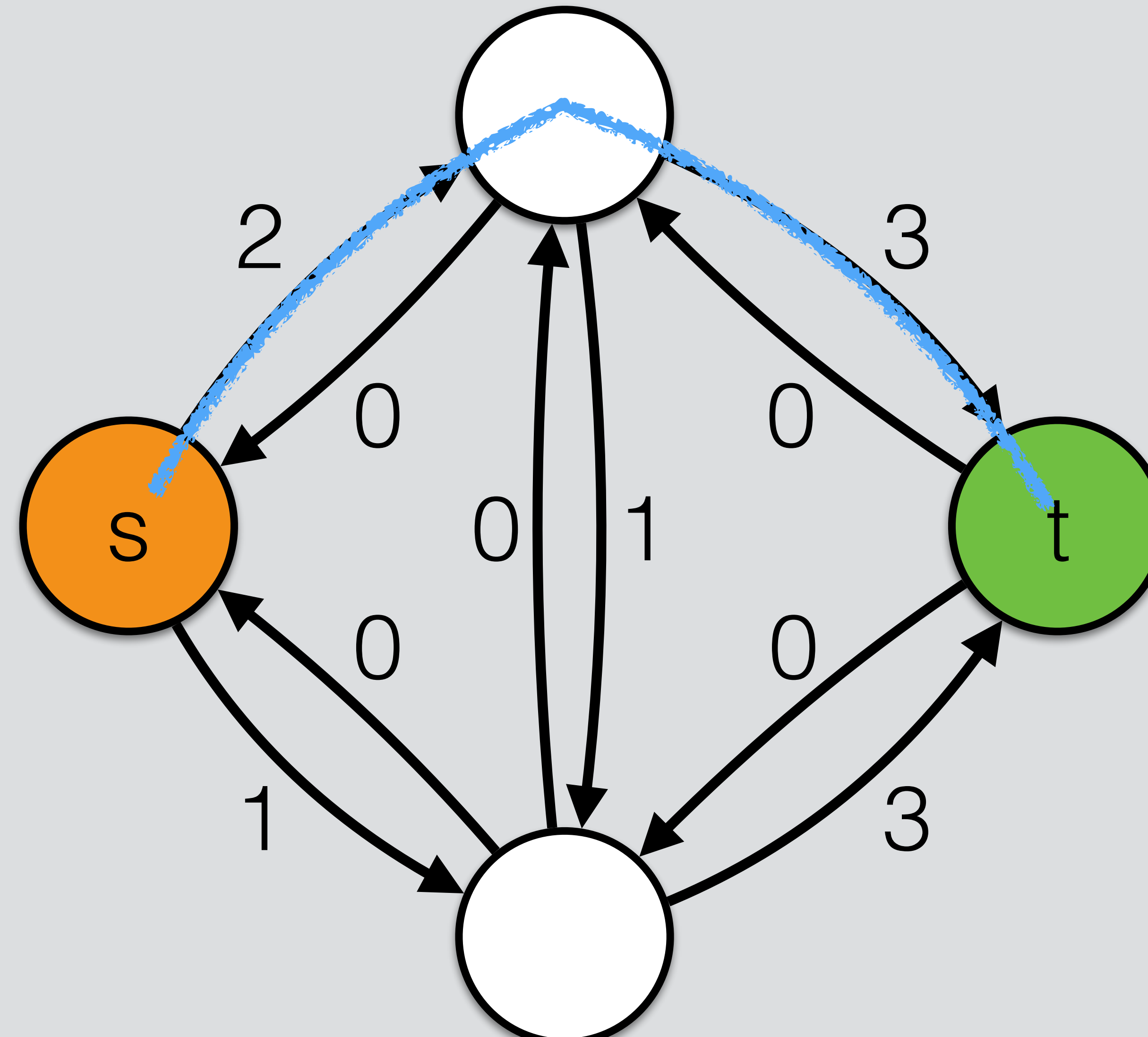
Residual graph



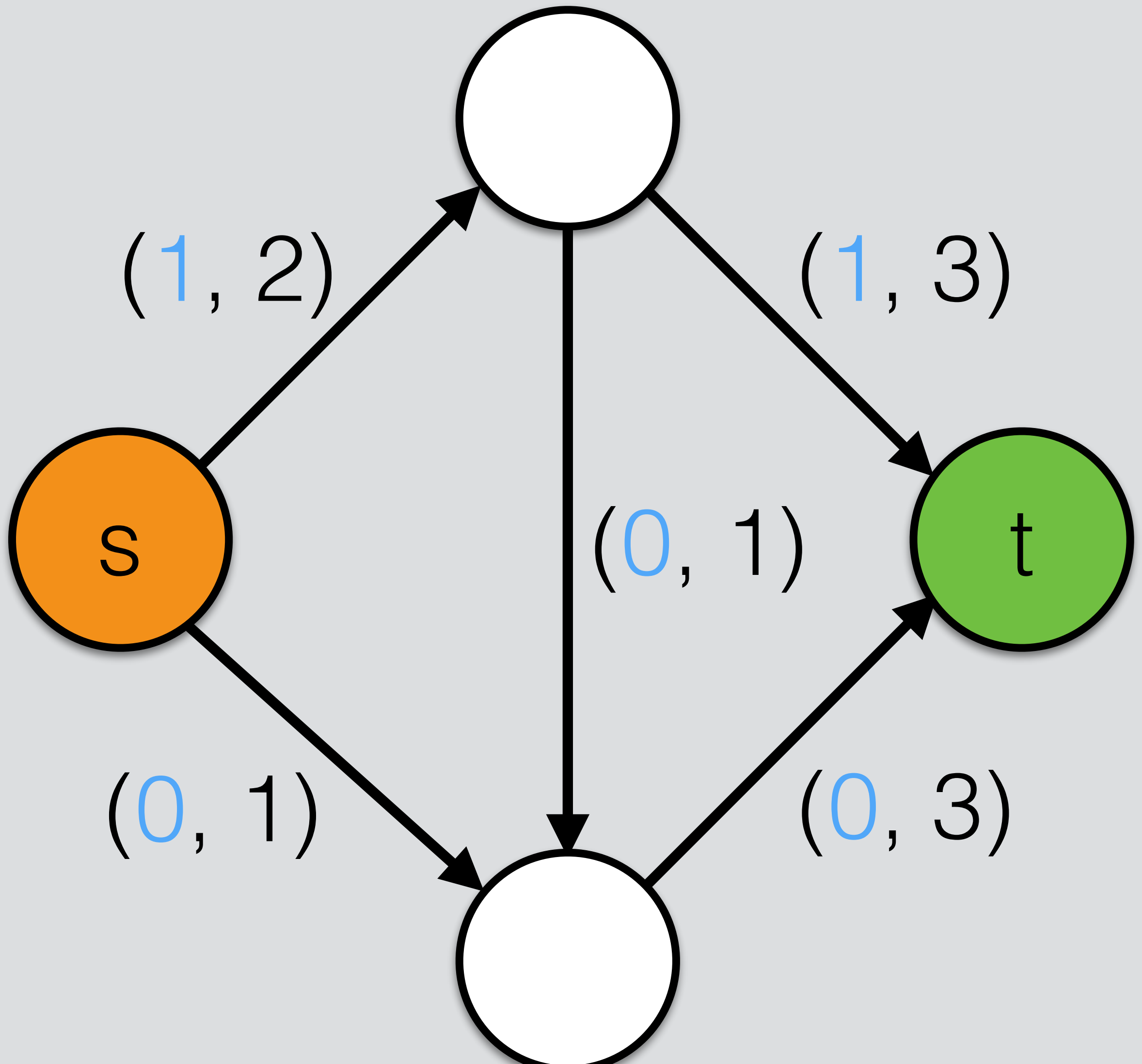
Graph (with flow)



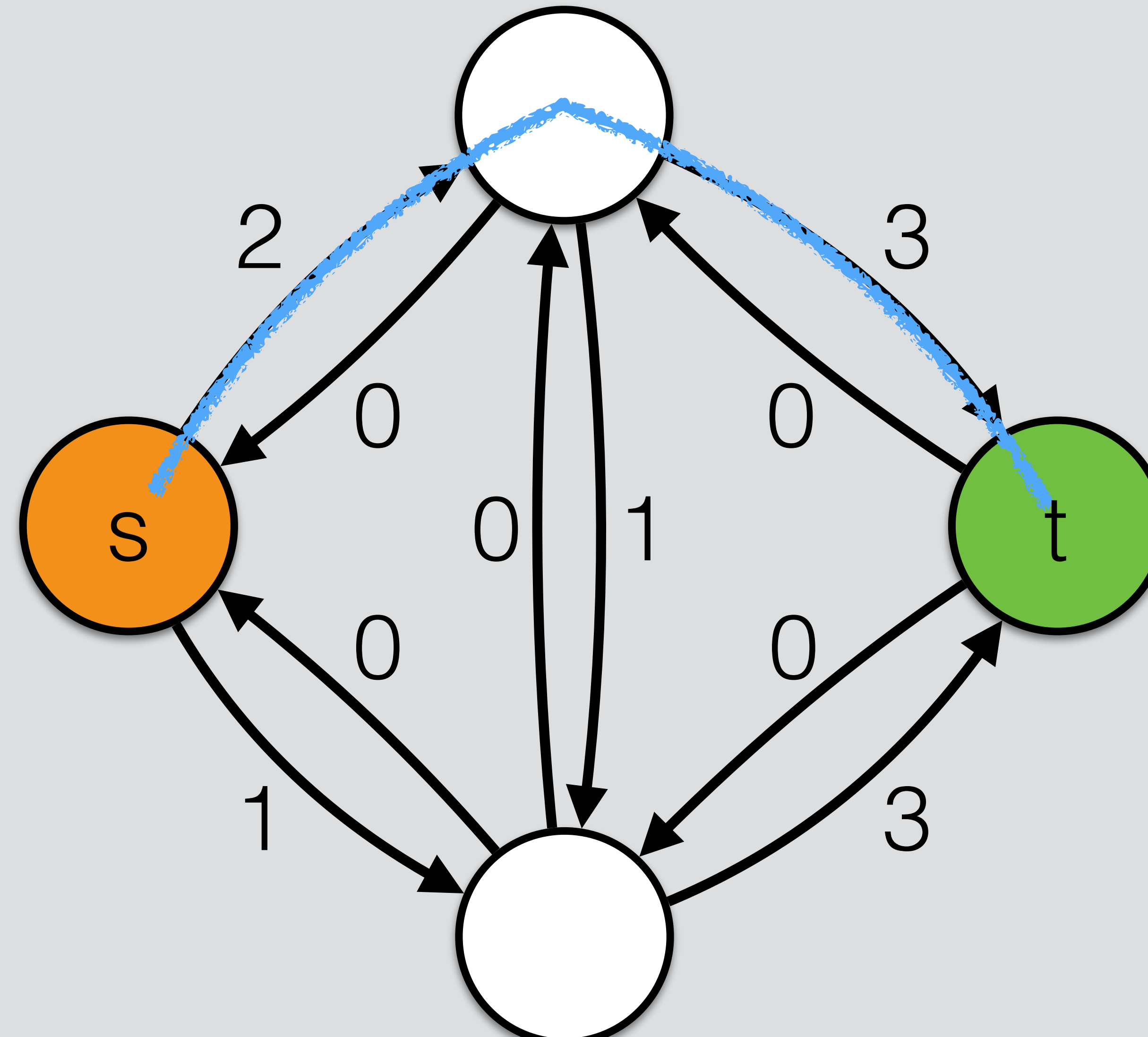
Residual graph



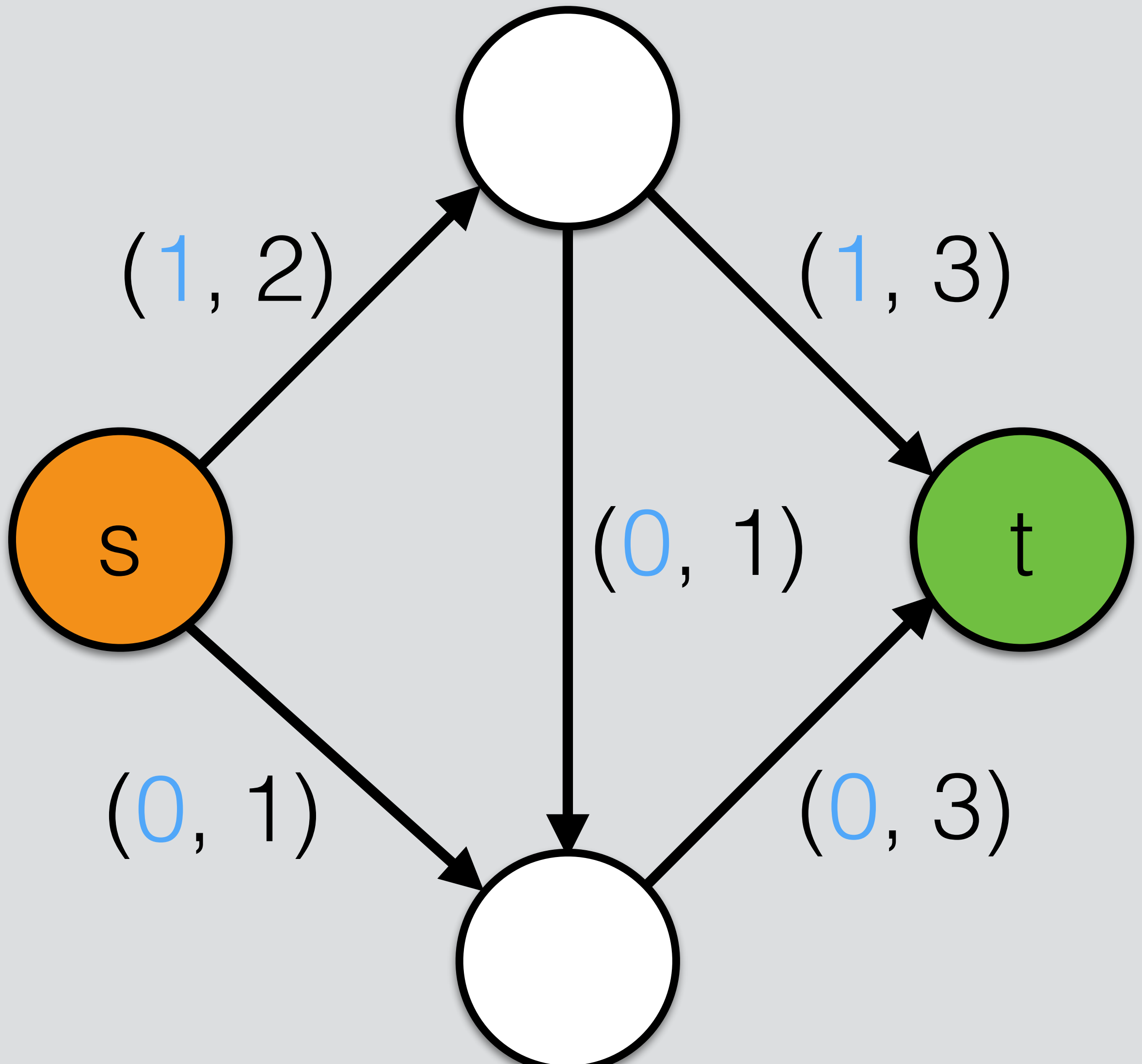
Graph (with flow)



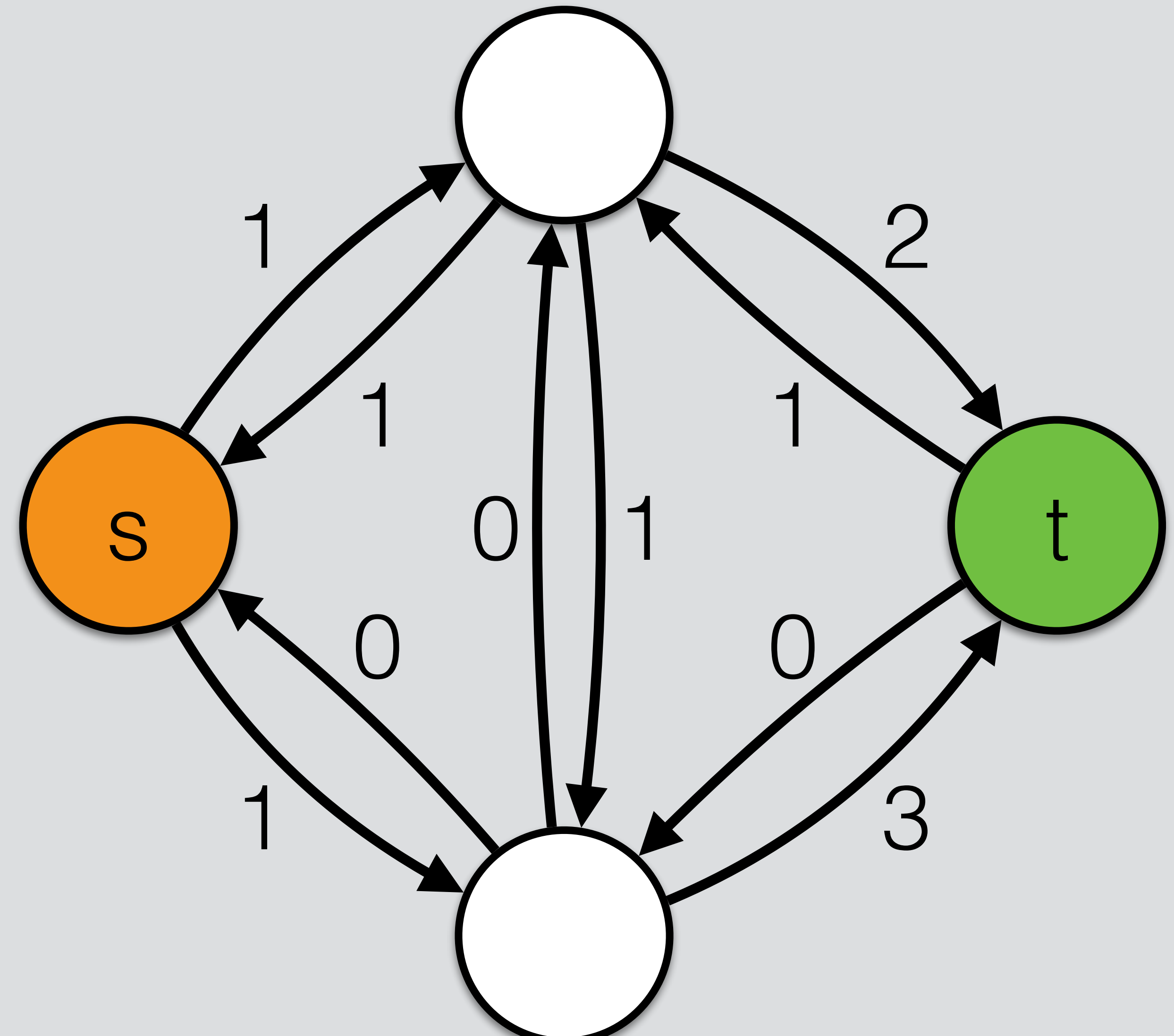
Residual graph



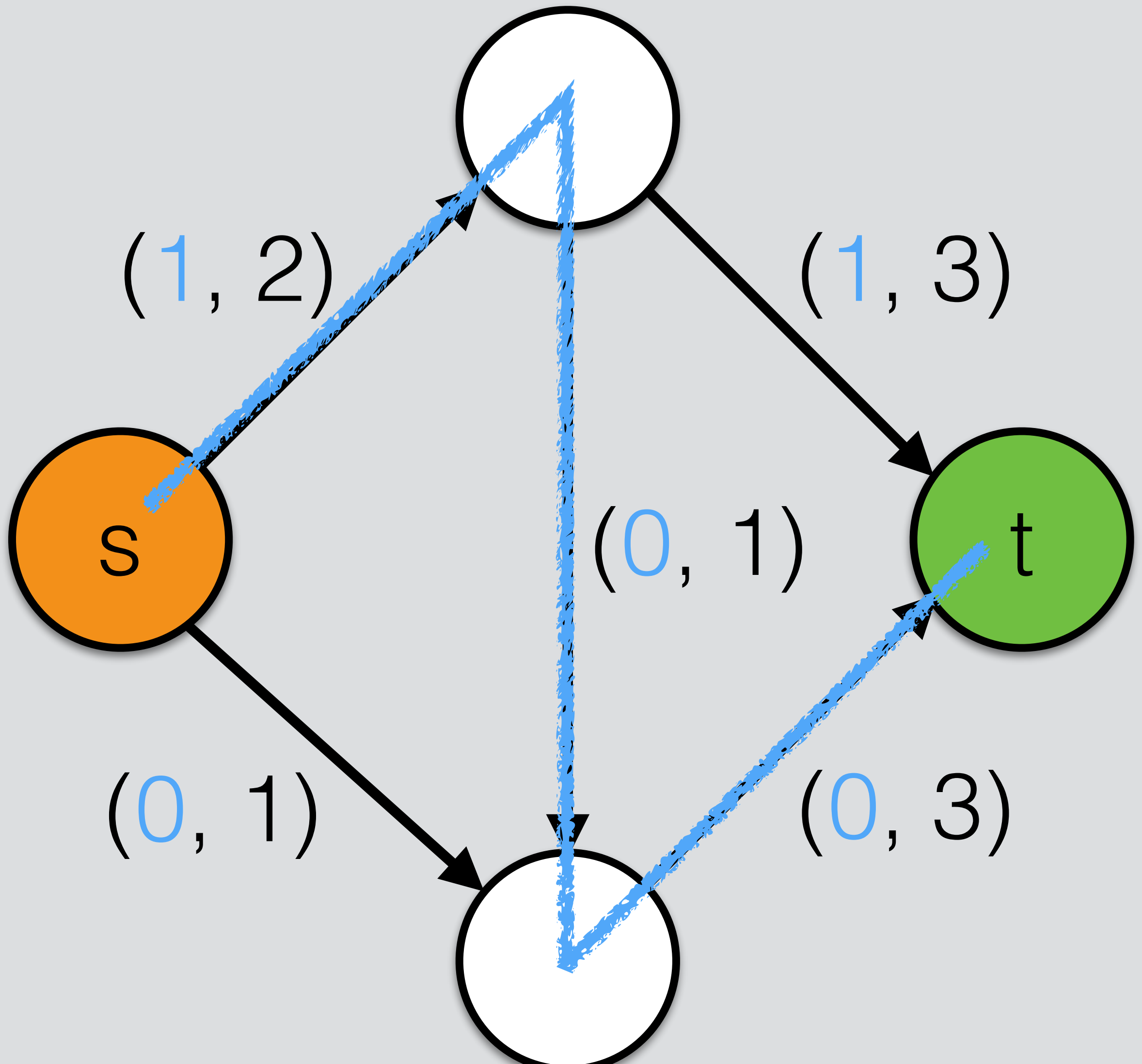
Graph (with flow)



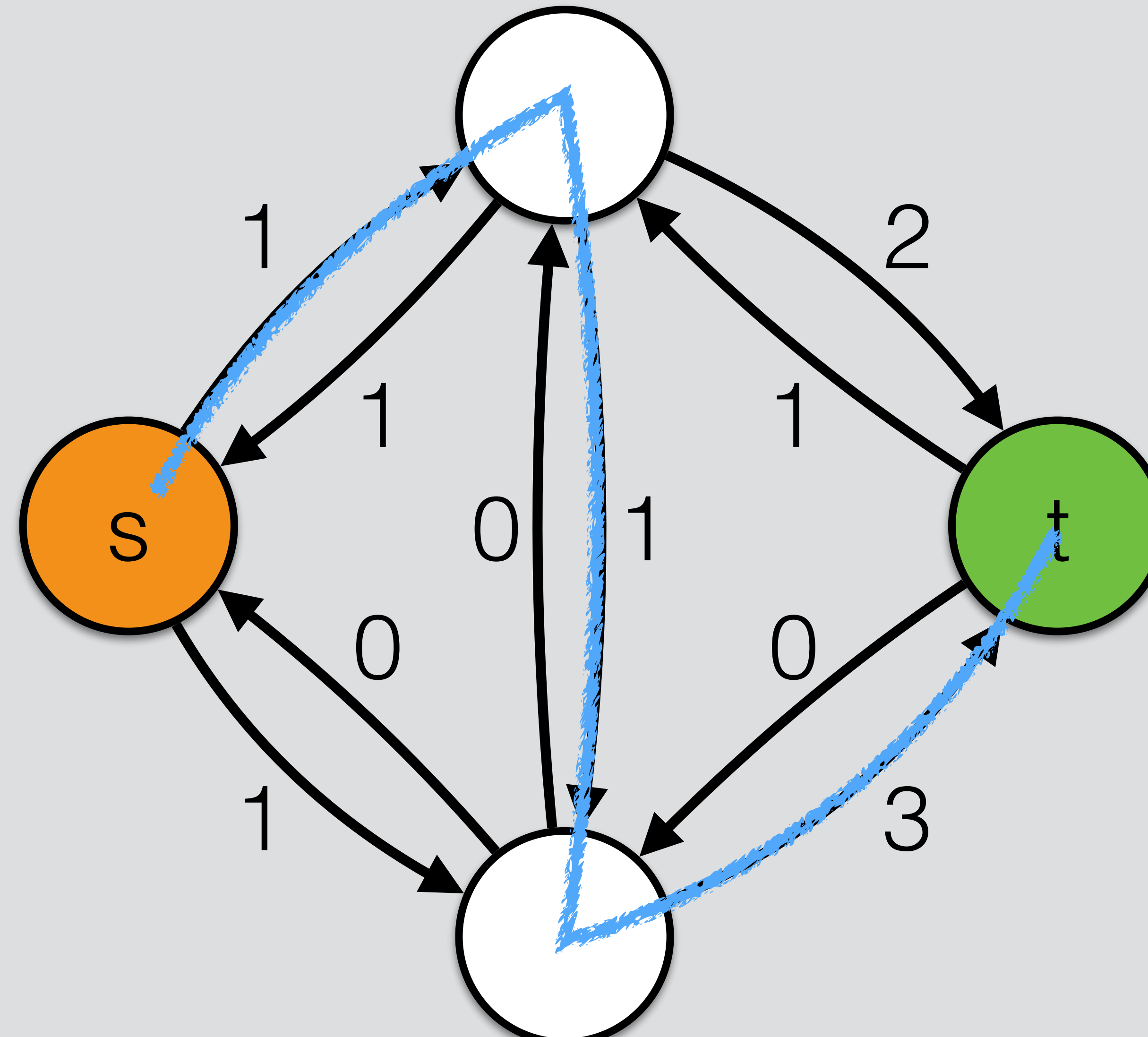
Residual graph



Graph (with flow)



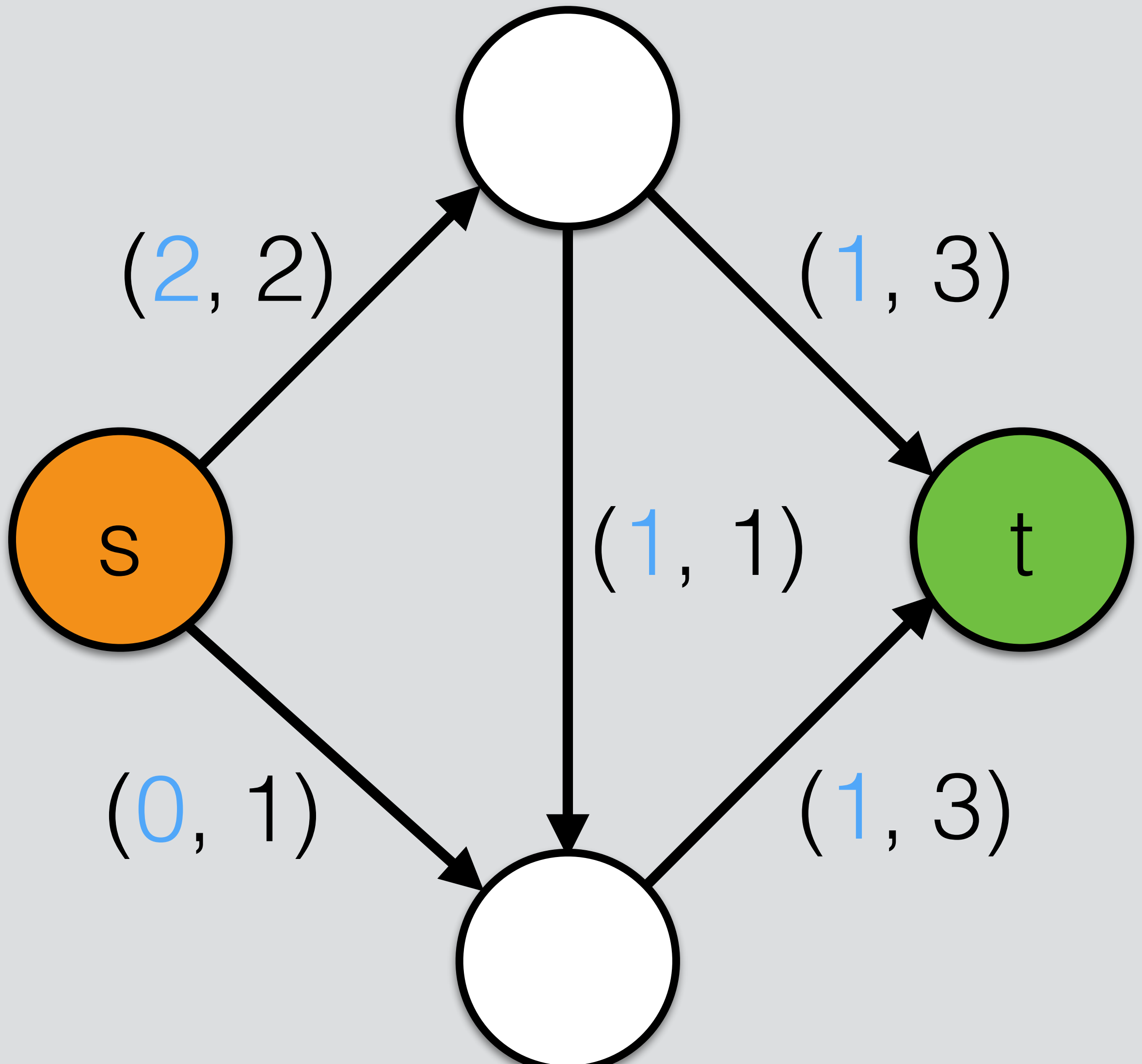
Residual graph



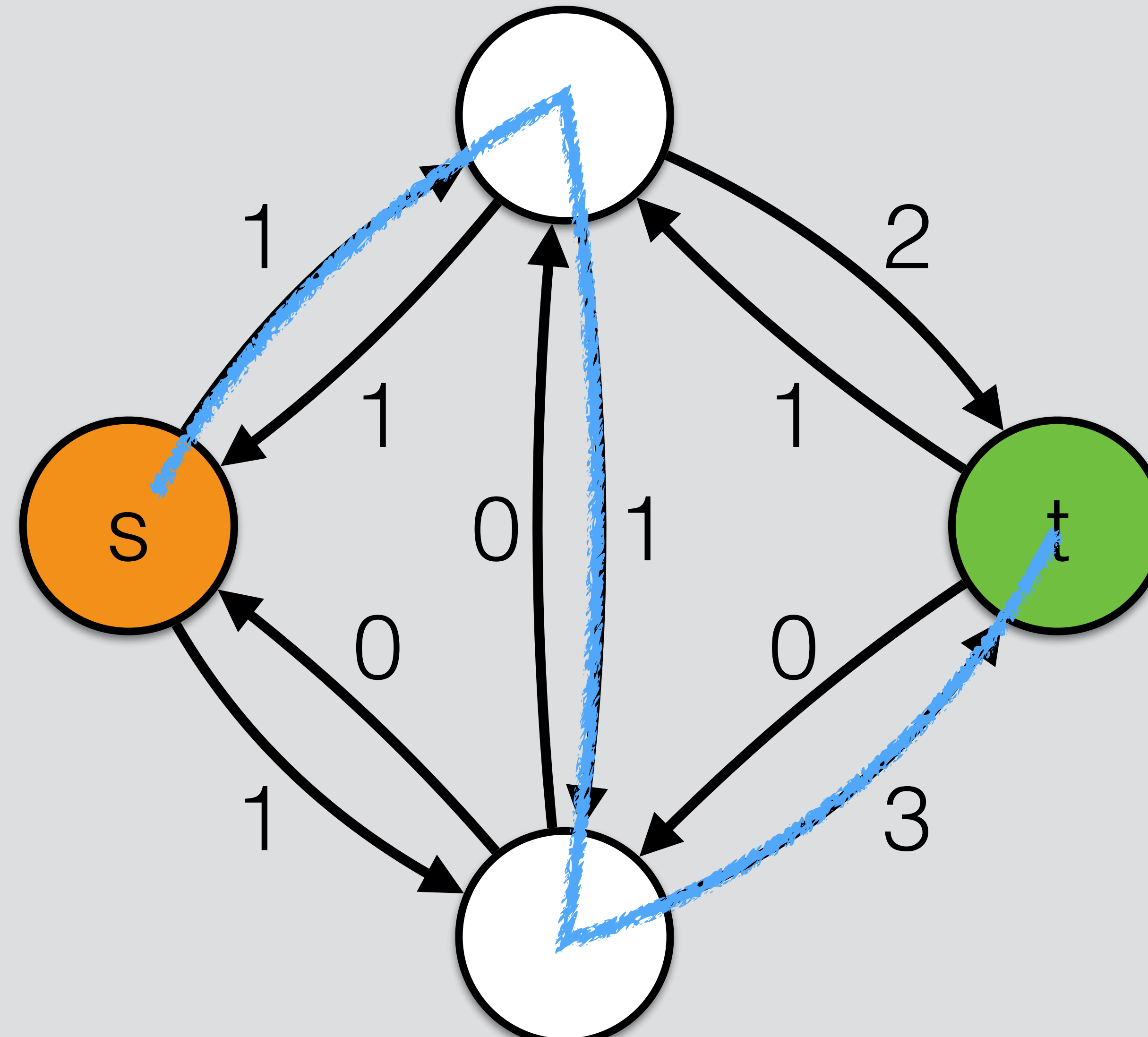
Augmenting path

(path from s to t along >0-weight edges)

Graph (with flow)

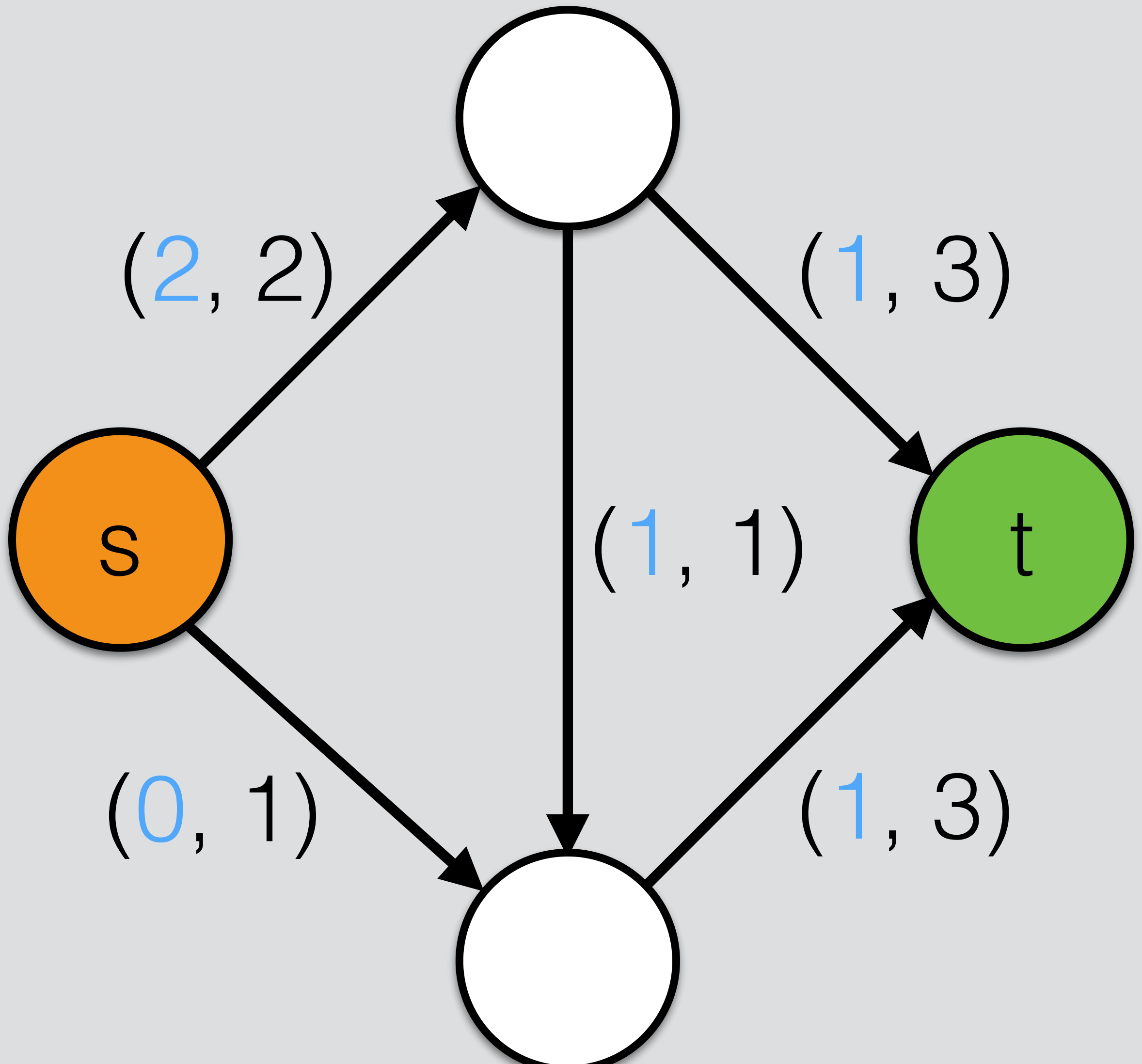


Residual graph

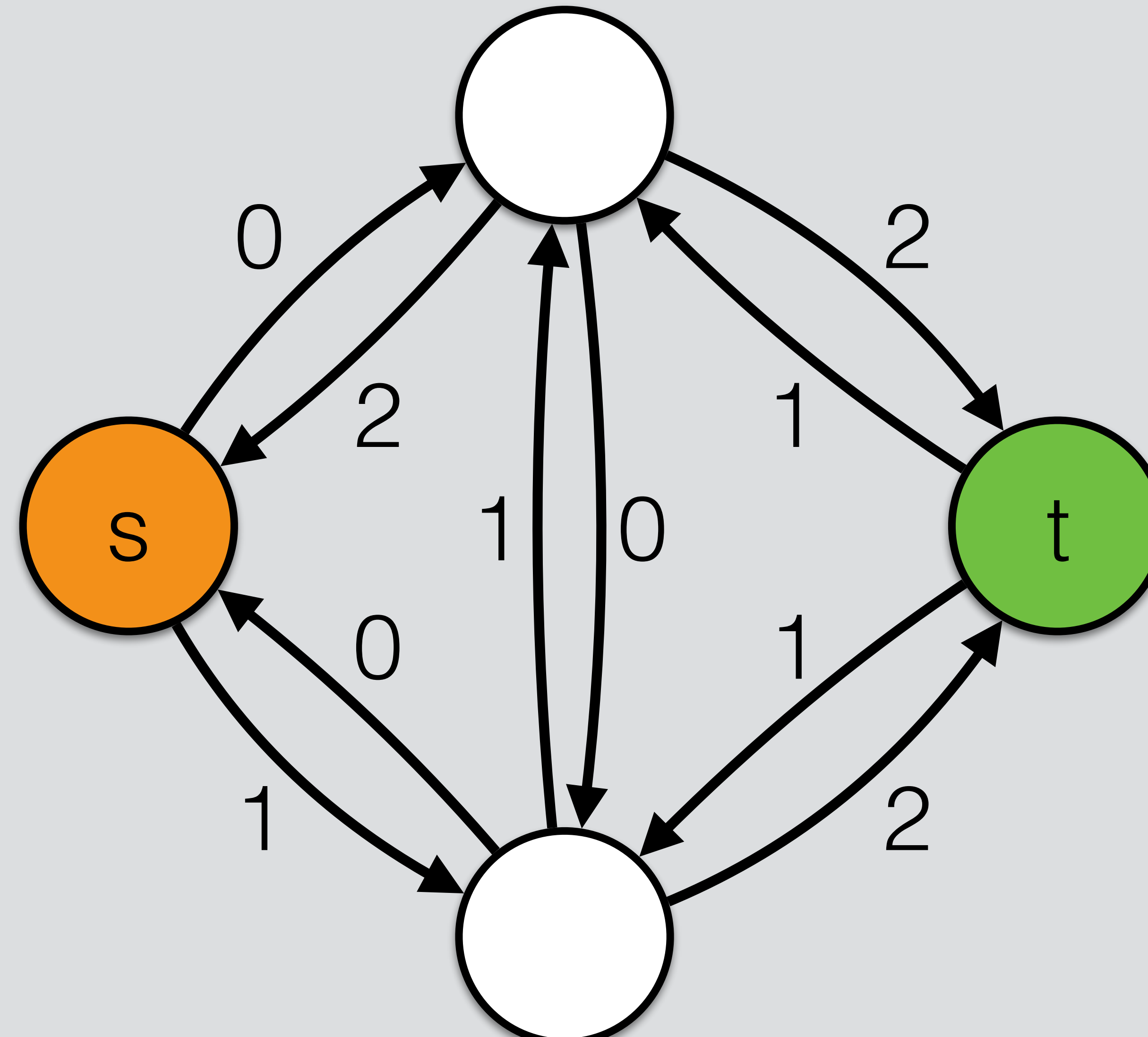


Augmenting path
(path from s to t along >0-weight edges)

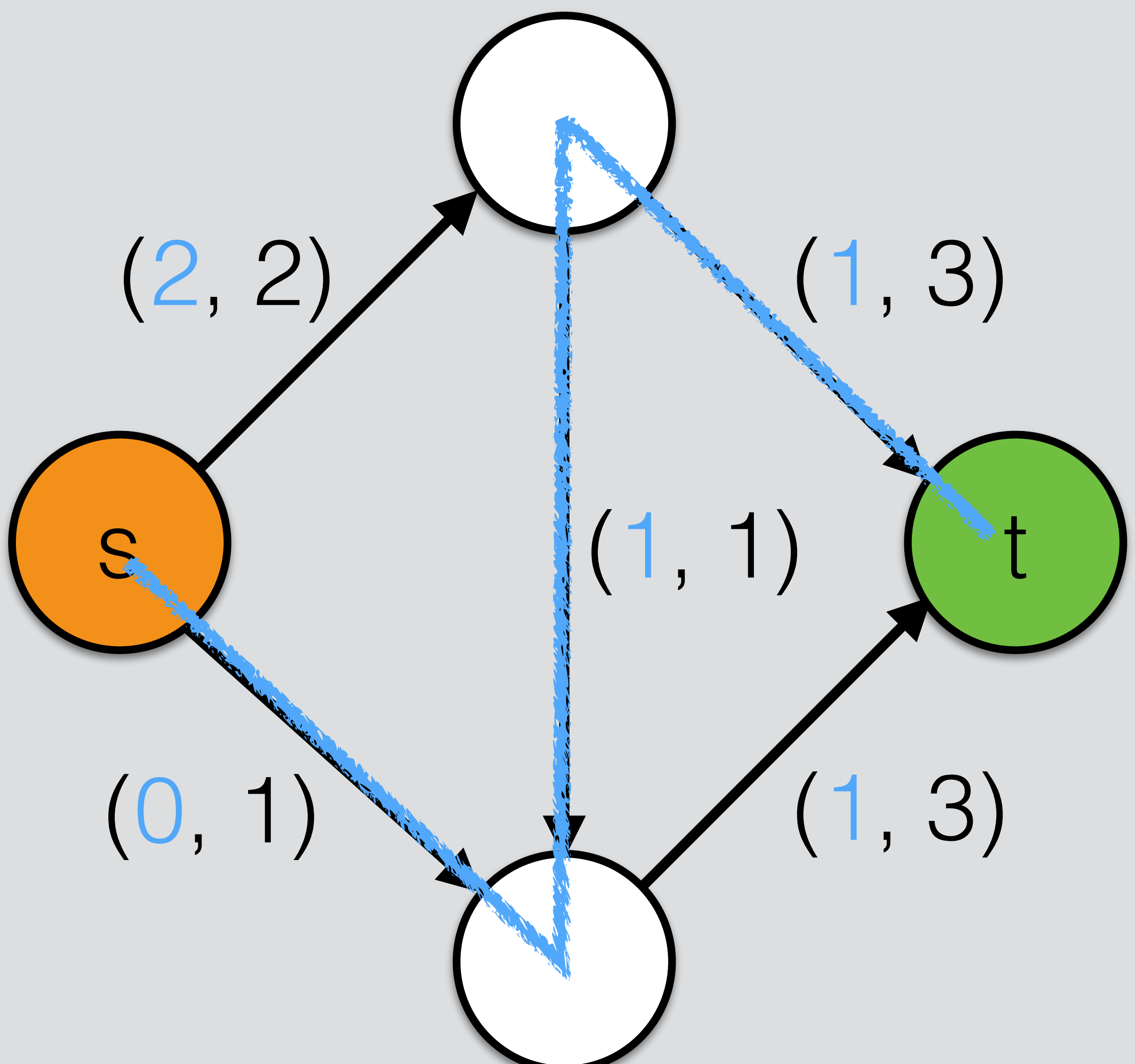
Graph (with flow)



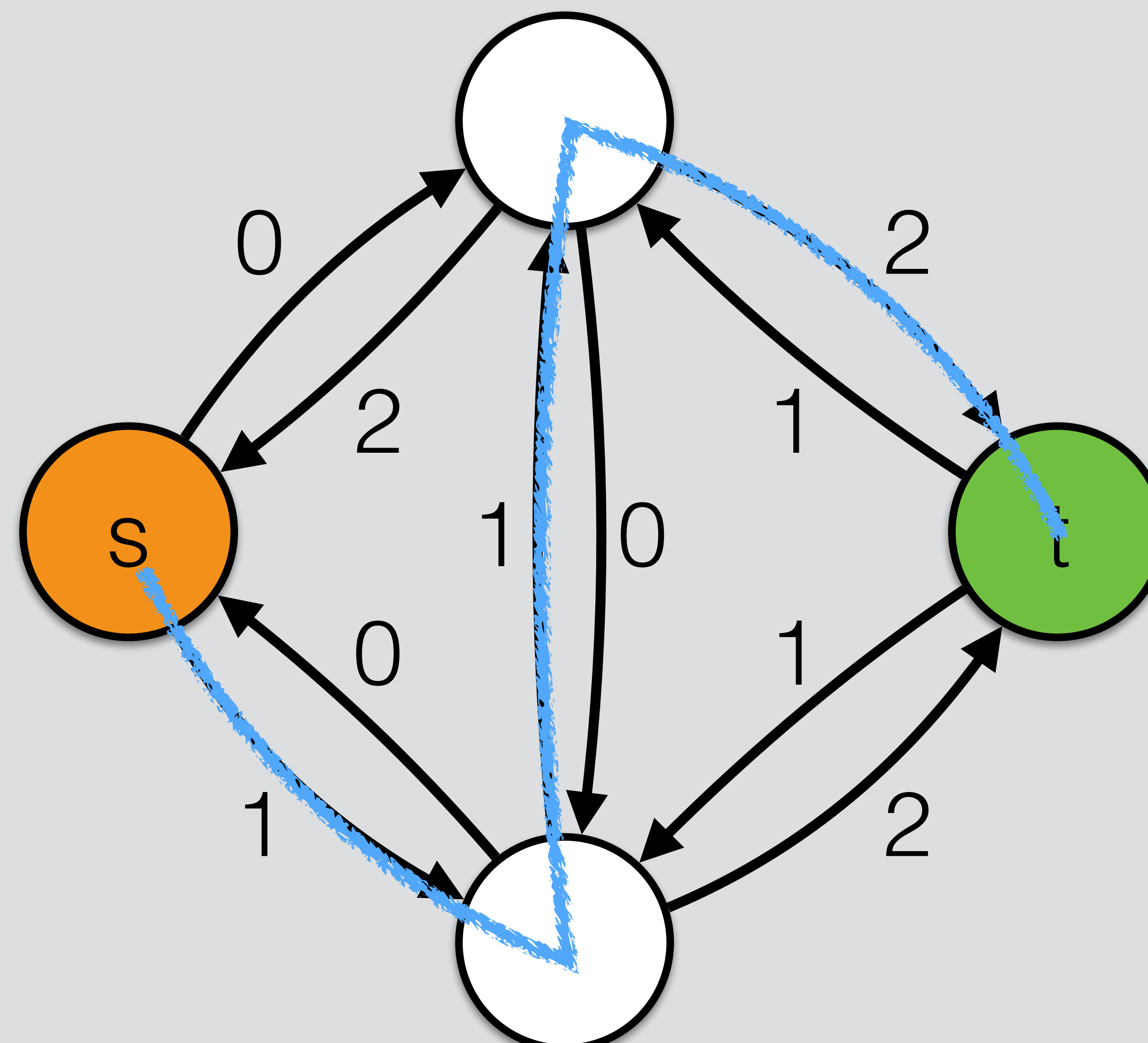
Residual graph



Graph (with flow)



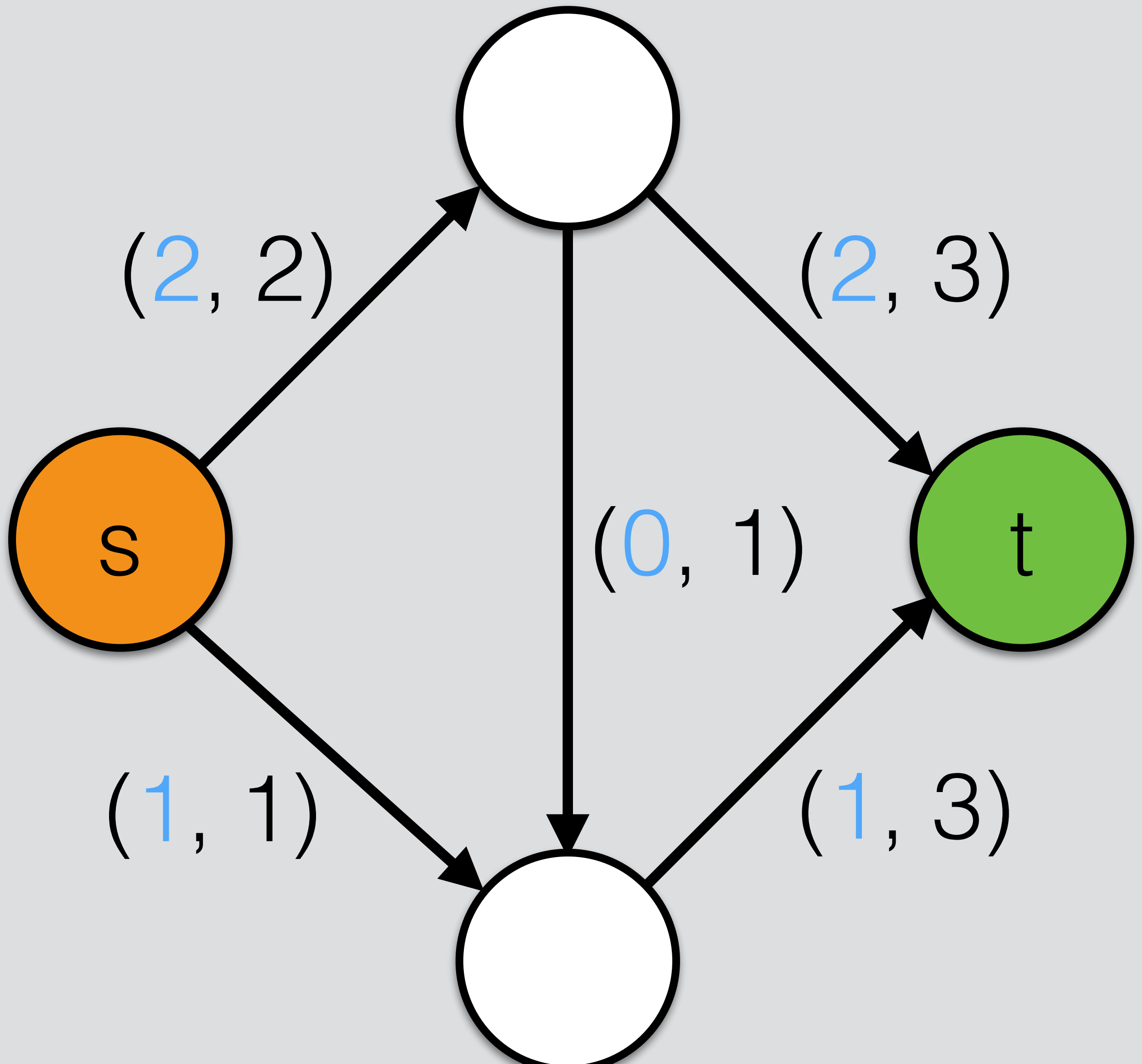
Residual graph



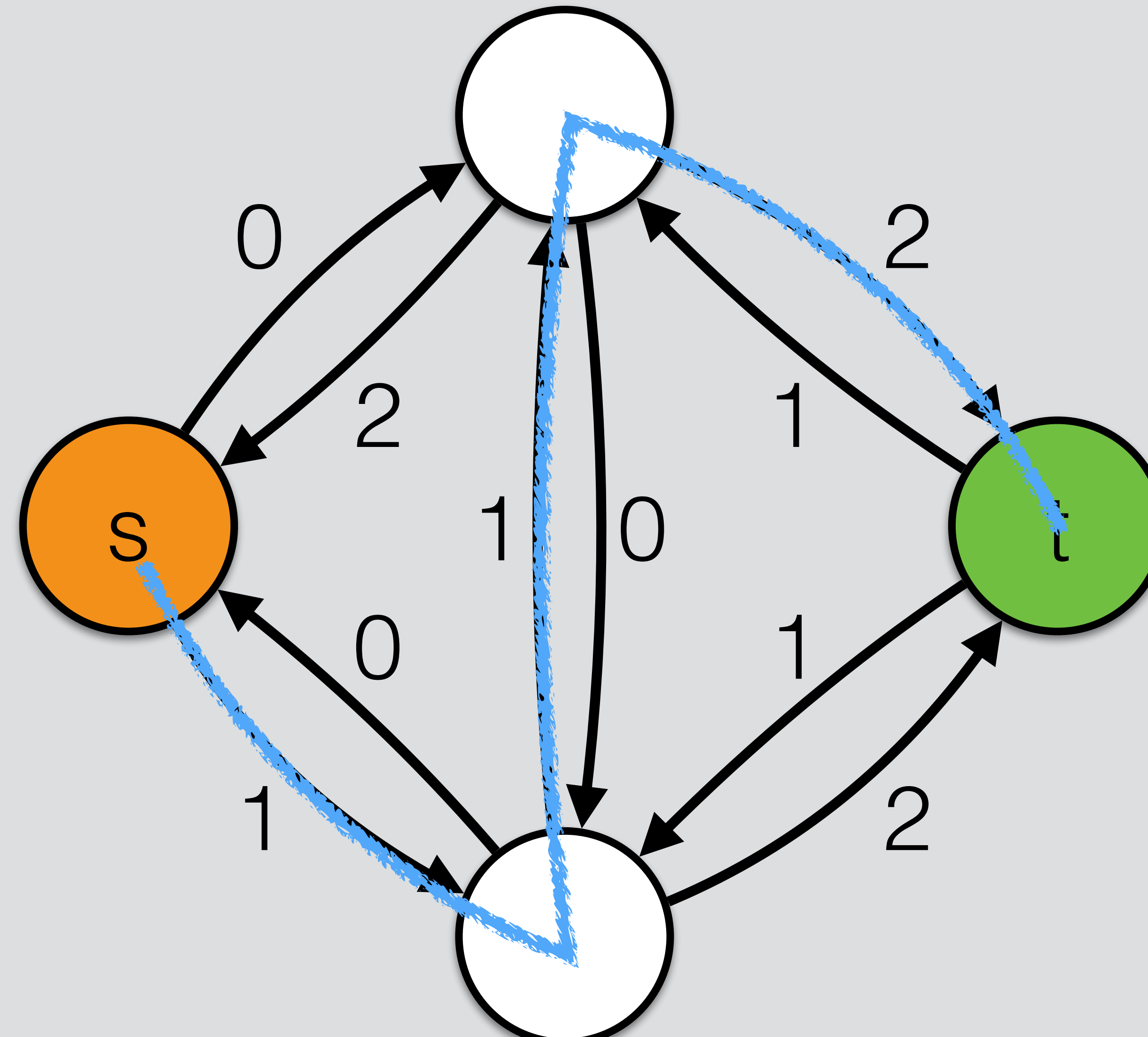
Augmenting path

(path from s to t along >0-weight edges)

Graph (with flow)

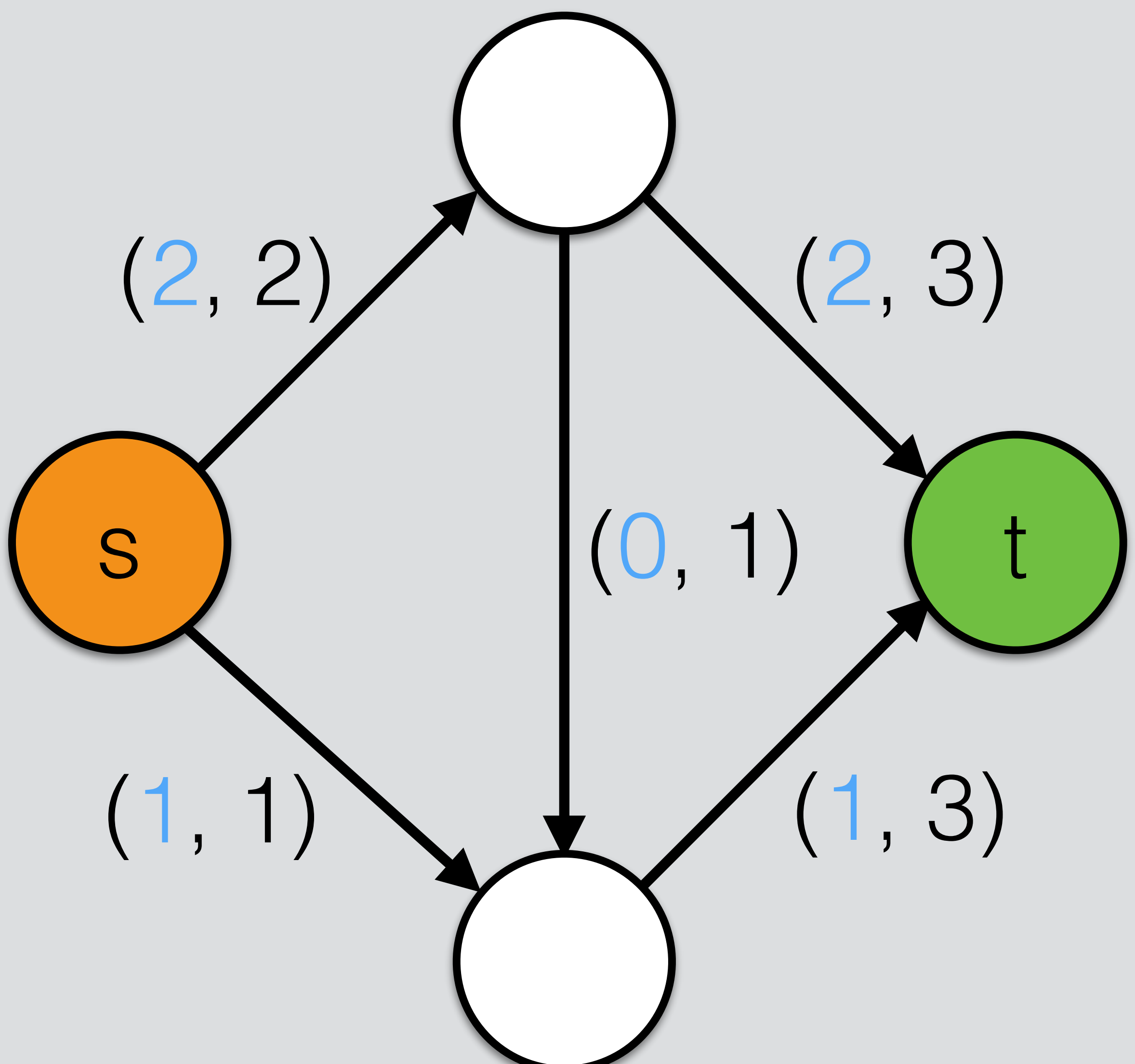


Residual graph

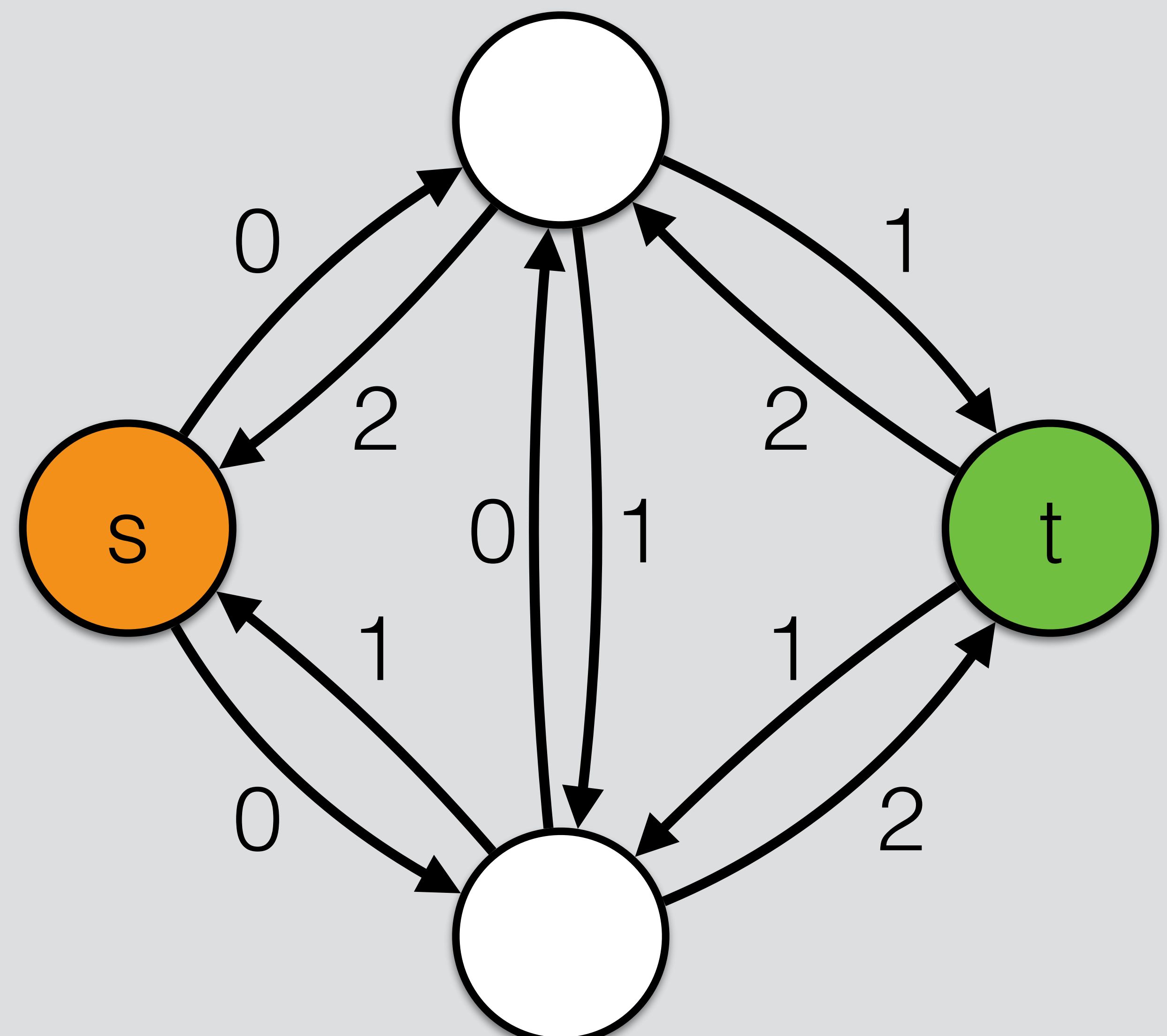


Augmenting path
(path from s to t along
>0-weight edges)

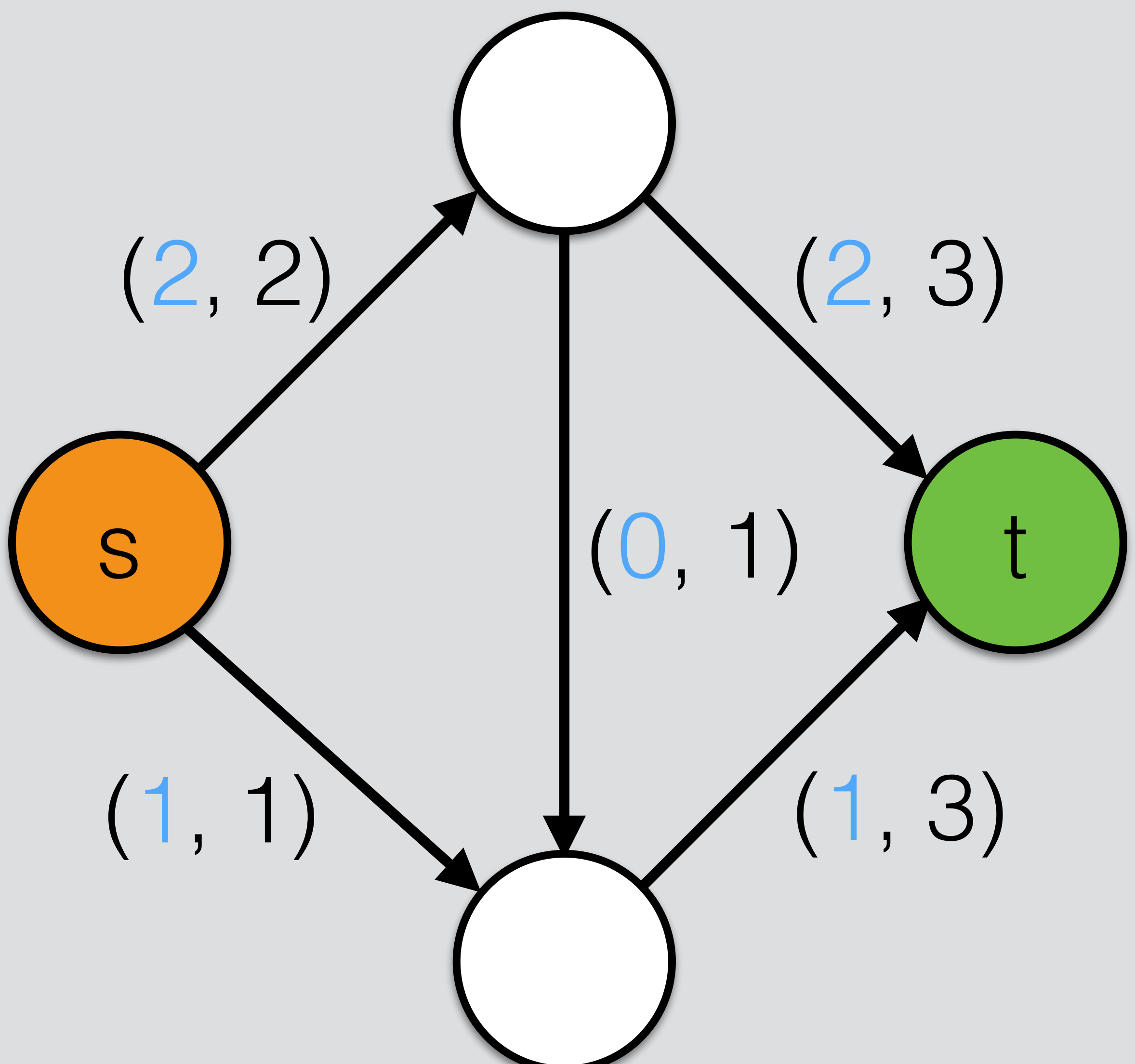
Graph (with flow)



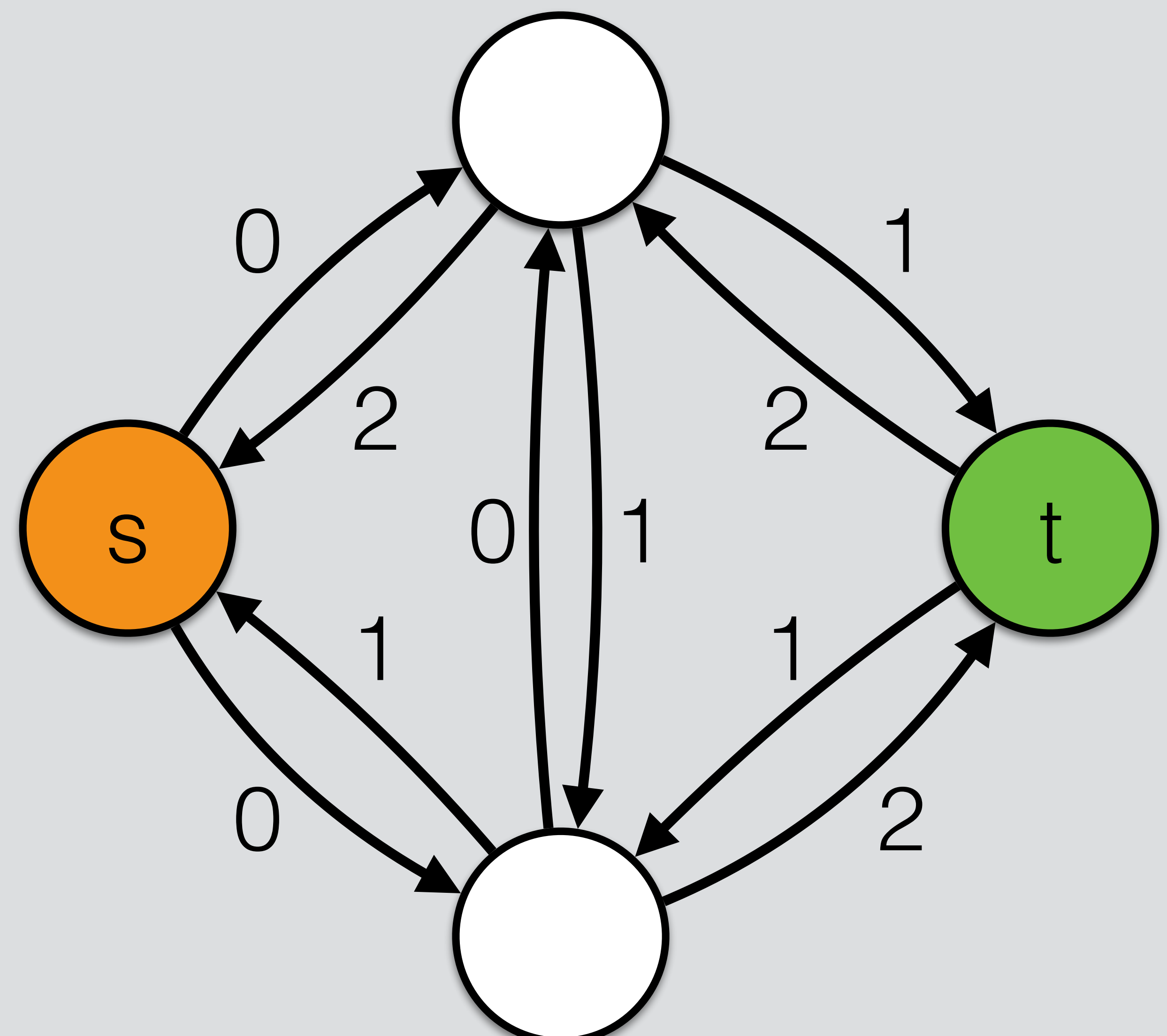
Residual graph



Graph (with flow)



Residual graph

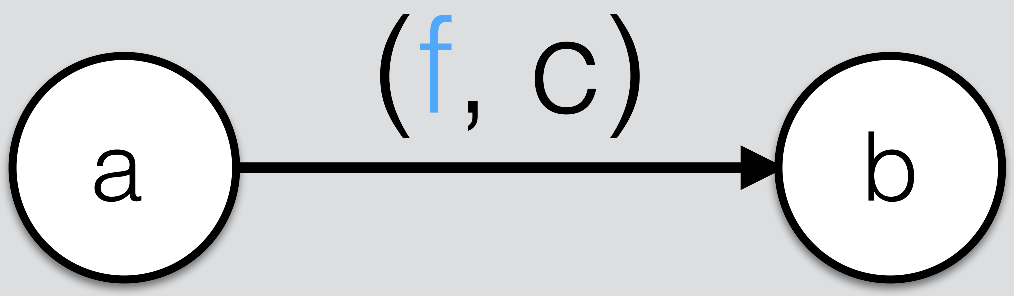


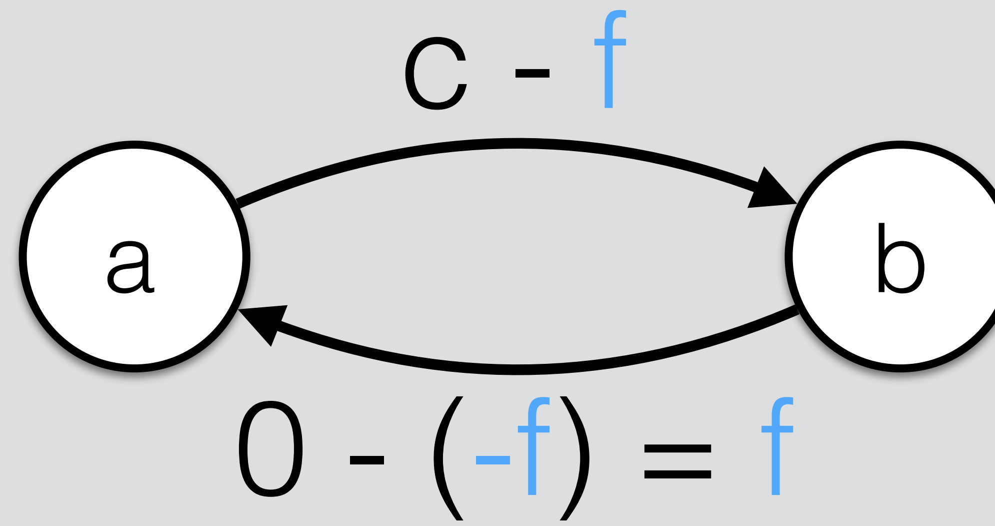
No augmenting path

(flow is maximum)

Edges (with flow)

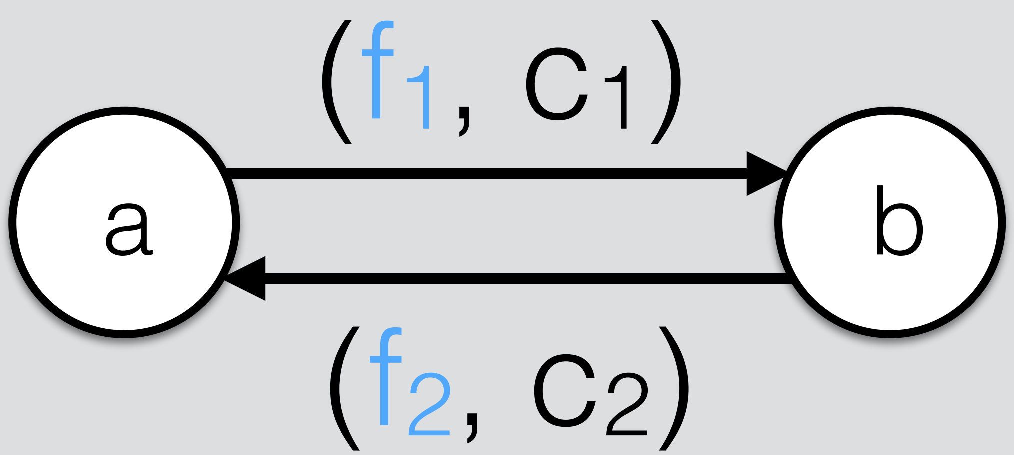
Residual edges

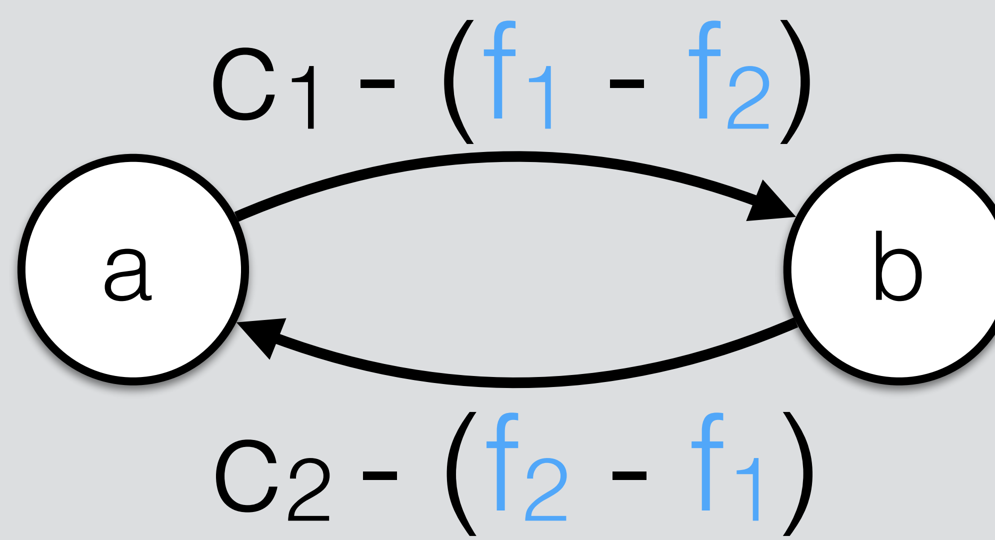




Net flow $f(a, b) = f$

$f(b, a) = -f$

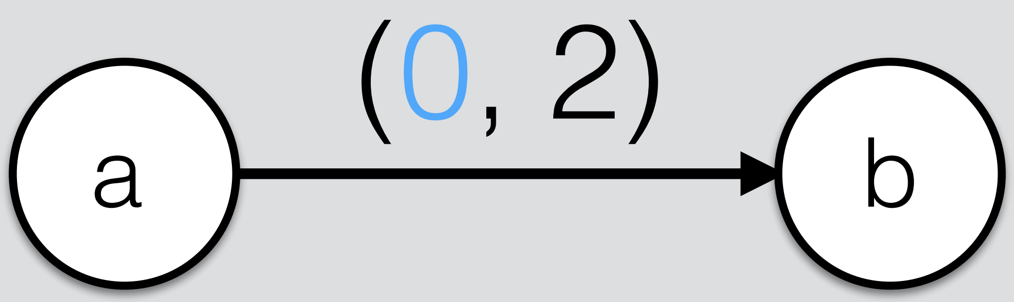




Net flow $f(a, b) = f_1 - f_2$

$f(b, a) = f_2 - f_1$

Edges (with flow)



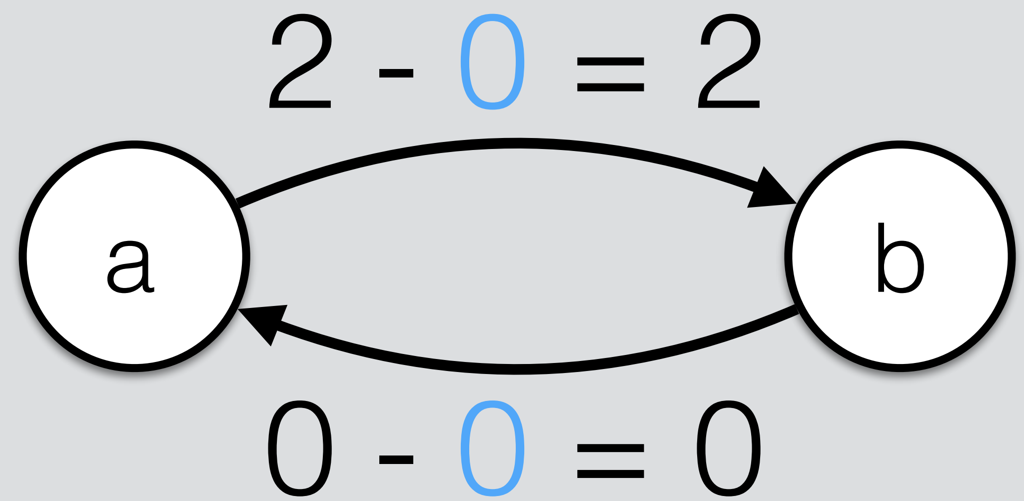
Net flow $f(a, b) = 0$

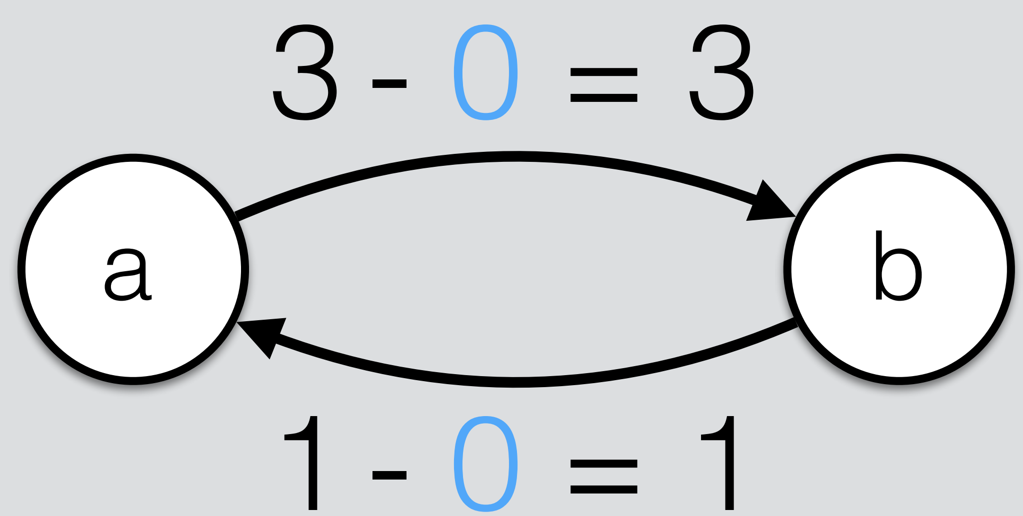
$f(b, a) = 0$

Net flow $f(a, b) = 0$

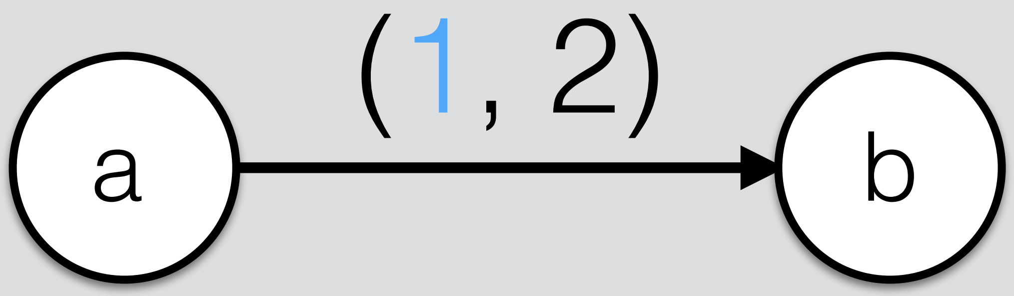
$f(b, a) = 0$

Residual edges





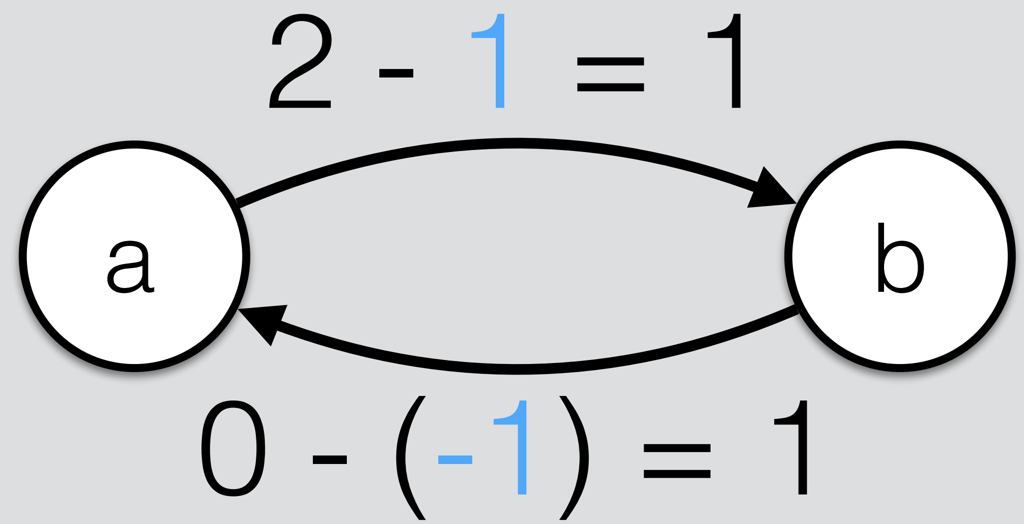
Edges (with flow)

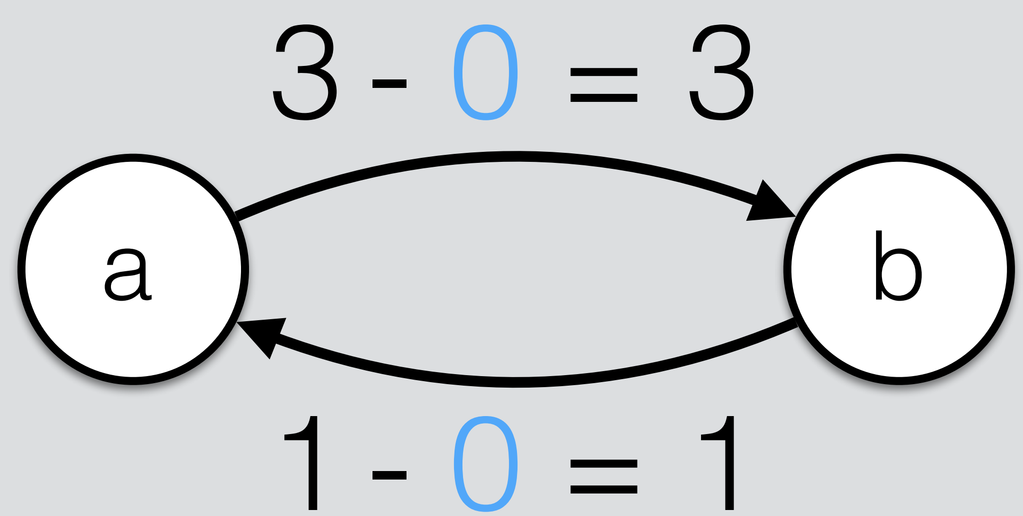


Net flow f(a, b) = 1
f(b, a) = -1

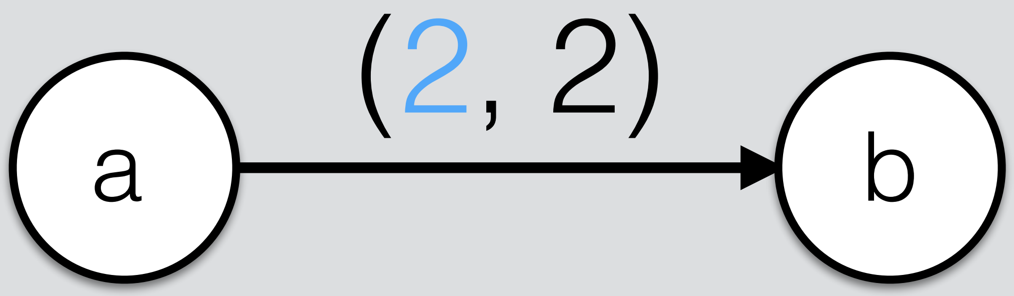
Net flow f(a, b) = 0
f(b, a) = 0

Residual edges





Edges (with flow)



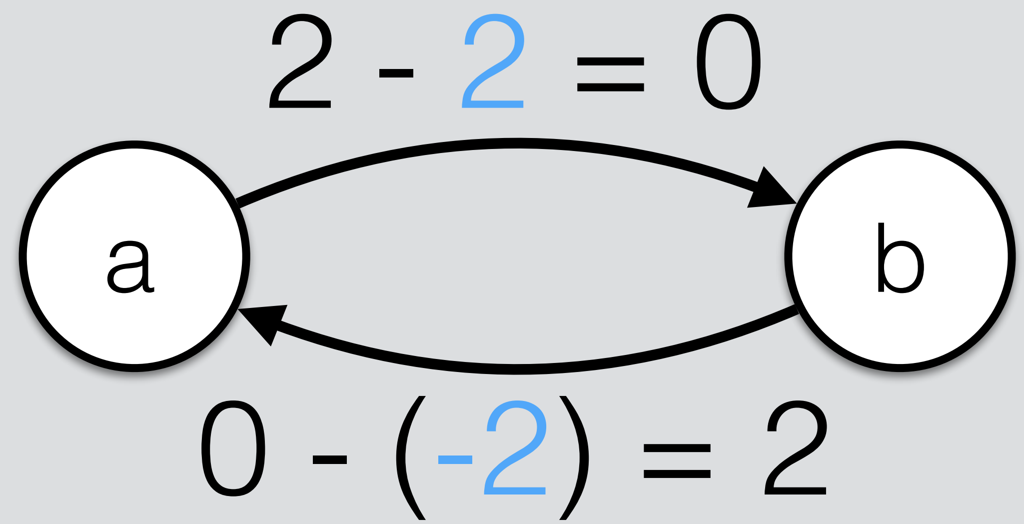
Net flow $f(a, b) = 2$

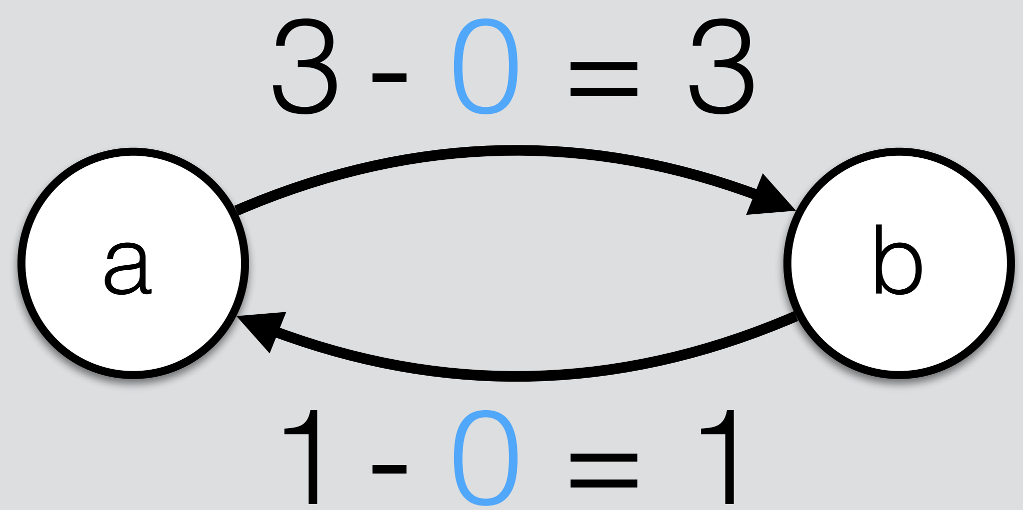
$f(b, a) = -2$

Net flow $f(a, b) = 0$

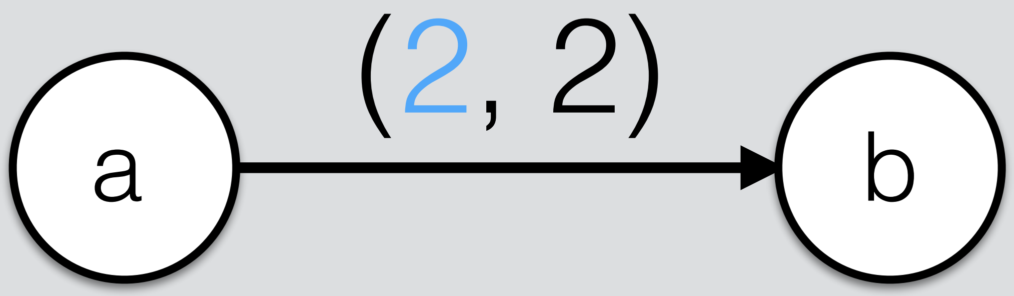
$f(b, a) = 0$

Residual edges





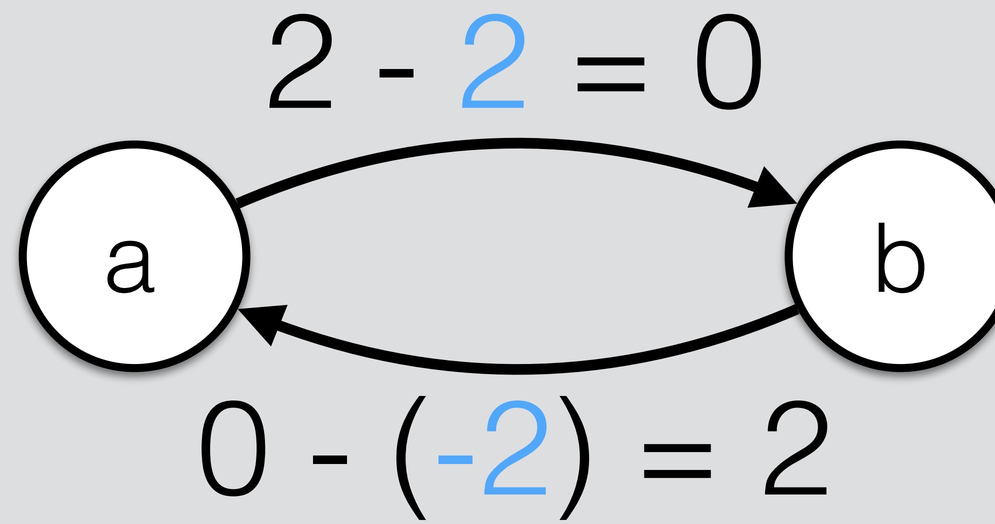
Edges (with flow)

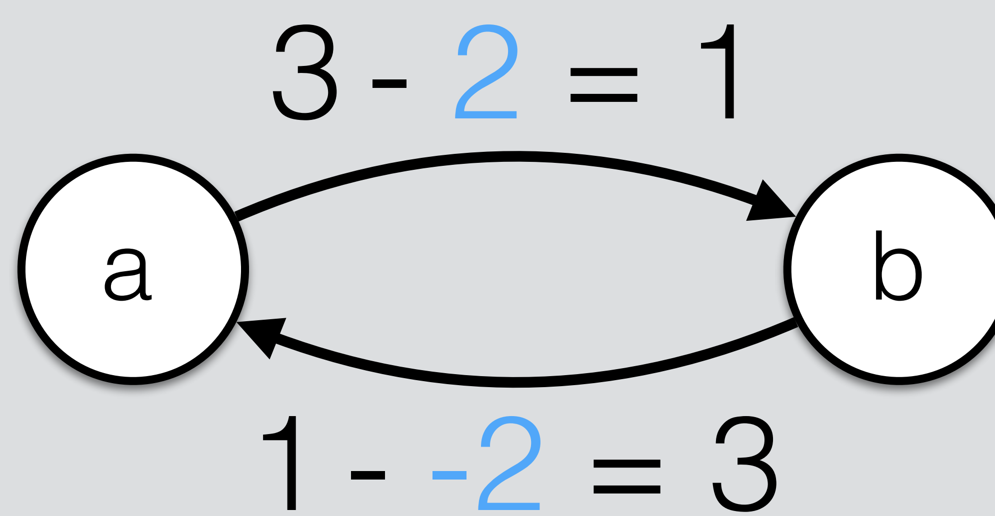


Net flow f(a, b) = 2
f(b, a) = -2

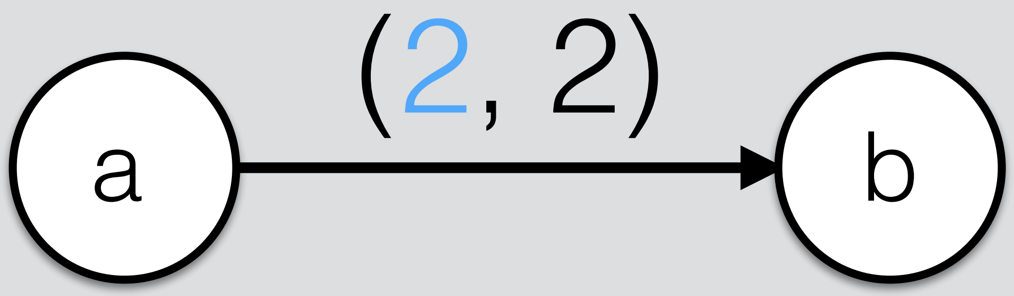
Net flow f(a, b) = 2
f(b, a) = -2

Residual edges





Edges (with flow)



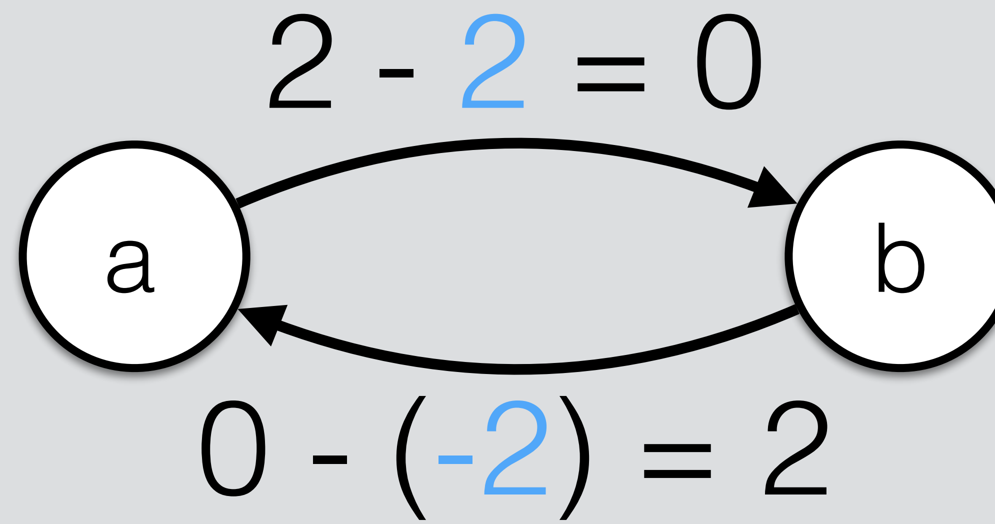
Net flow $f(a, b) = 2$

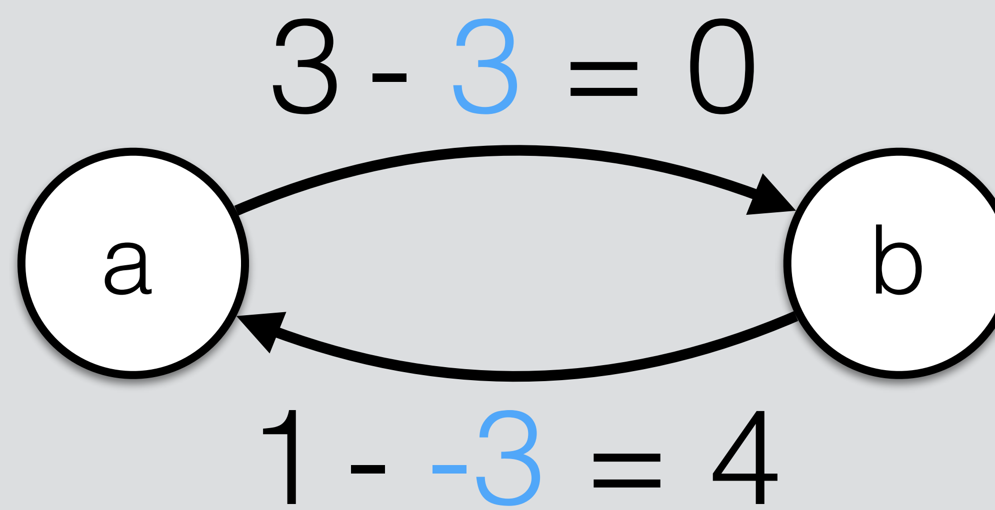
$f(b, a) = -2$

Net flow $f(a, b) = 3$

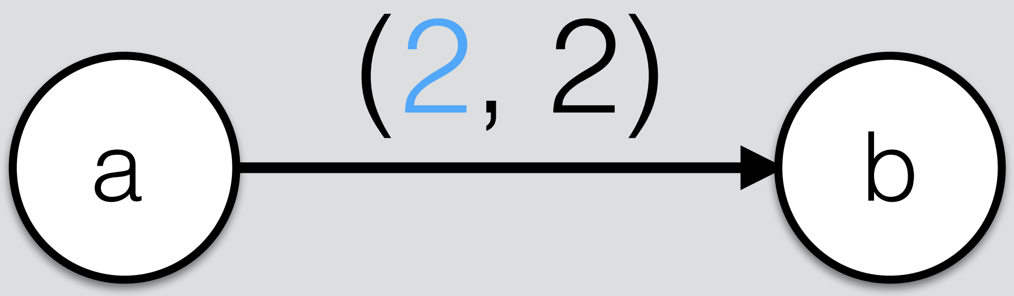
$f(b, a) = -3$

Residual edges





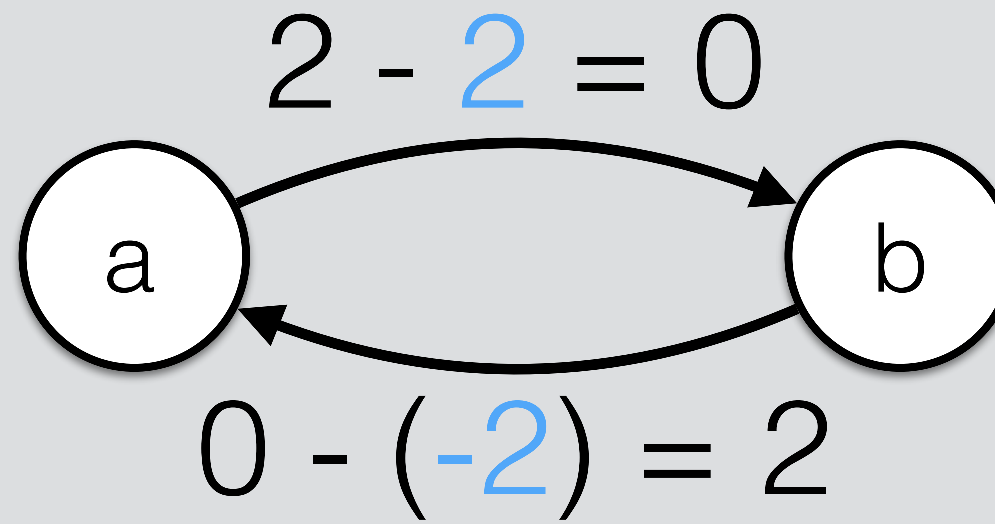
Edges (with flow)

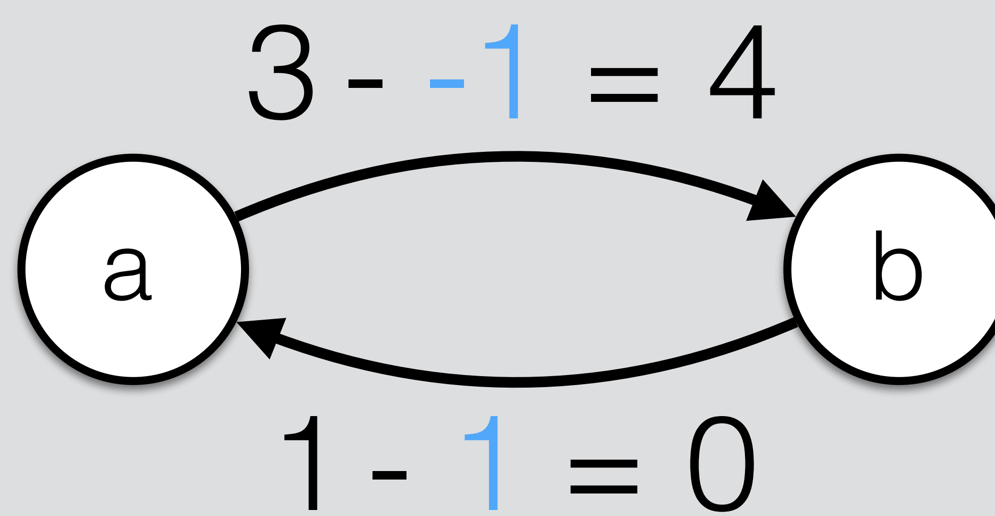


Net flow $f(a, b) = 2$
$f(b, a) = -2$

Net flow $f(a, b) = -1$
$f(b, a) = 1$

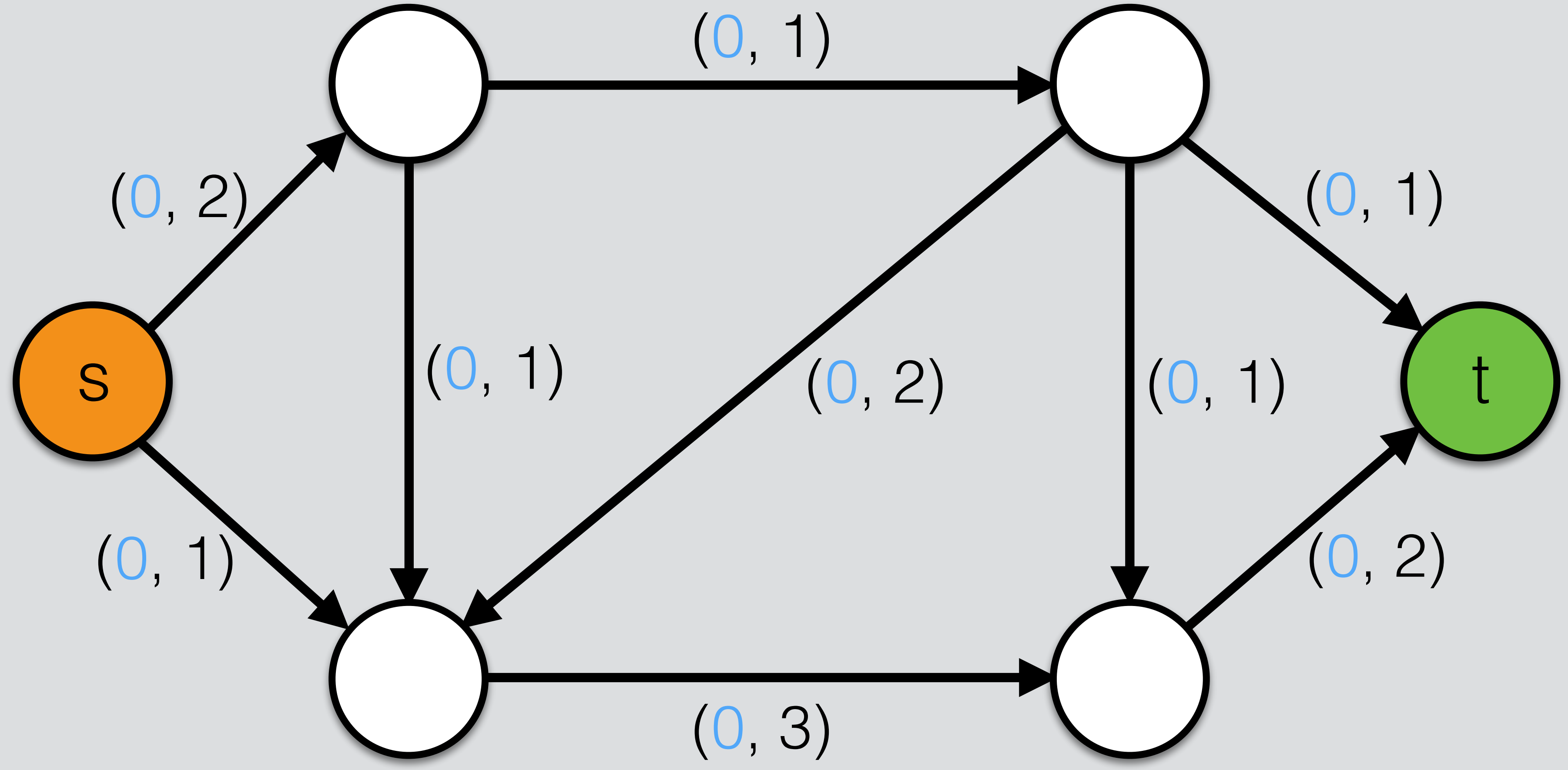
Residual edges



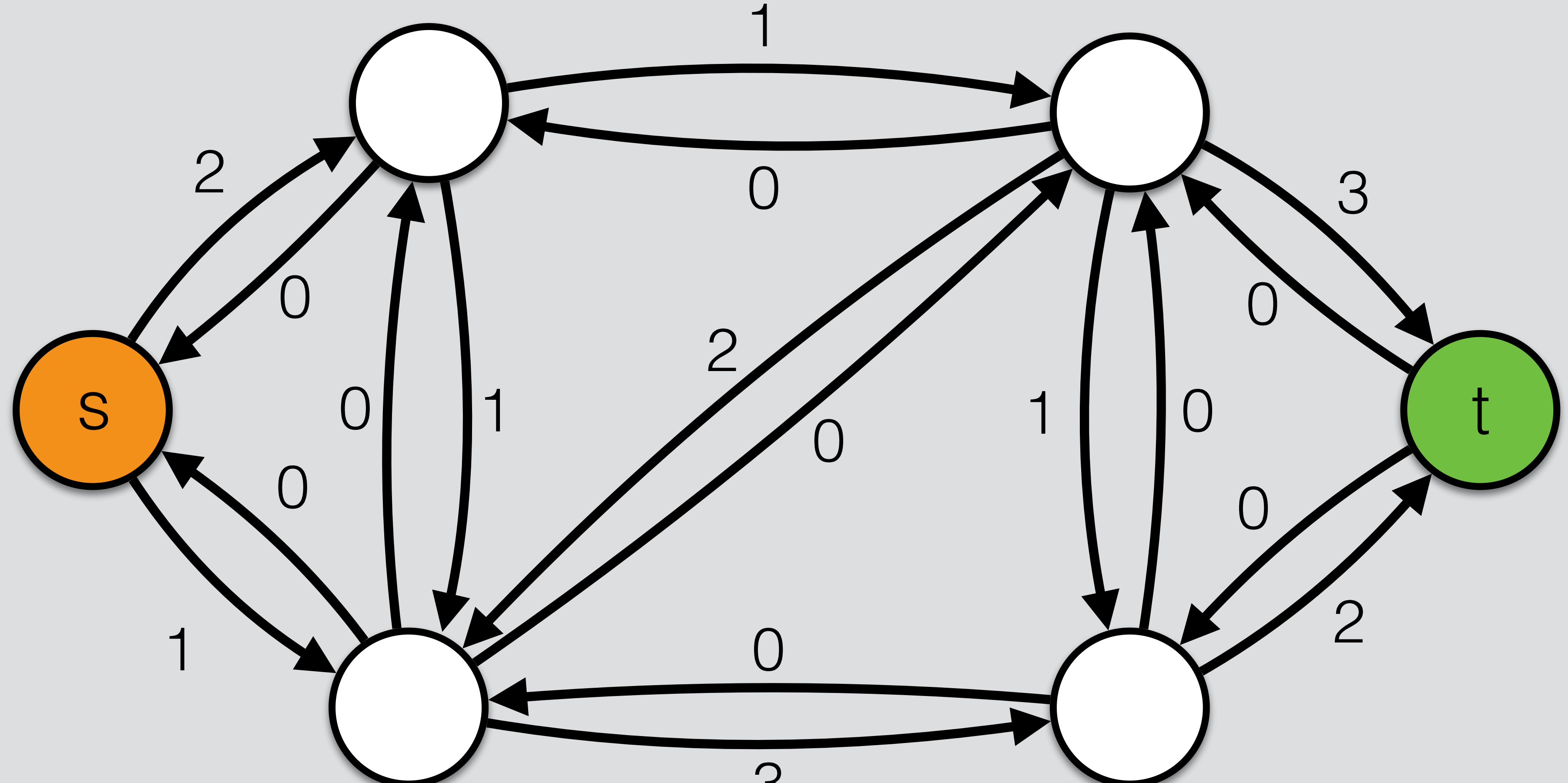


Ford-Fulkerson Maximum Flow Algorithm

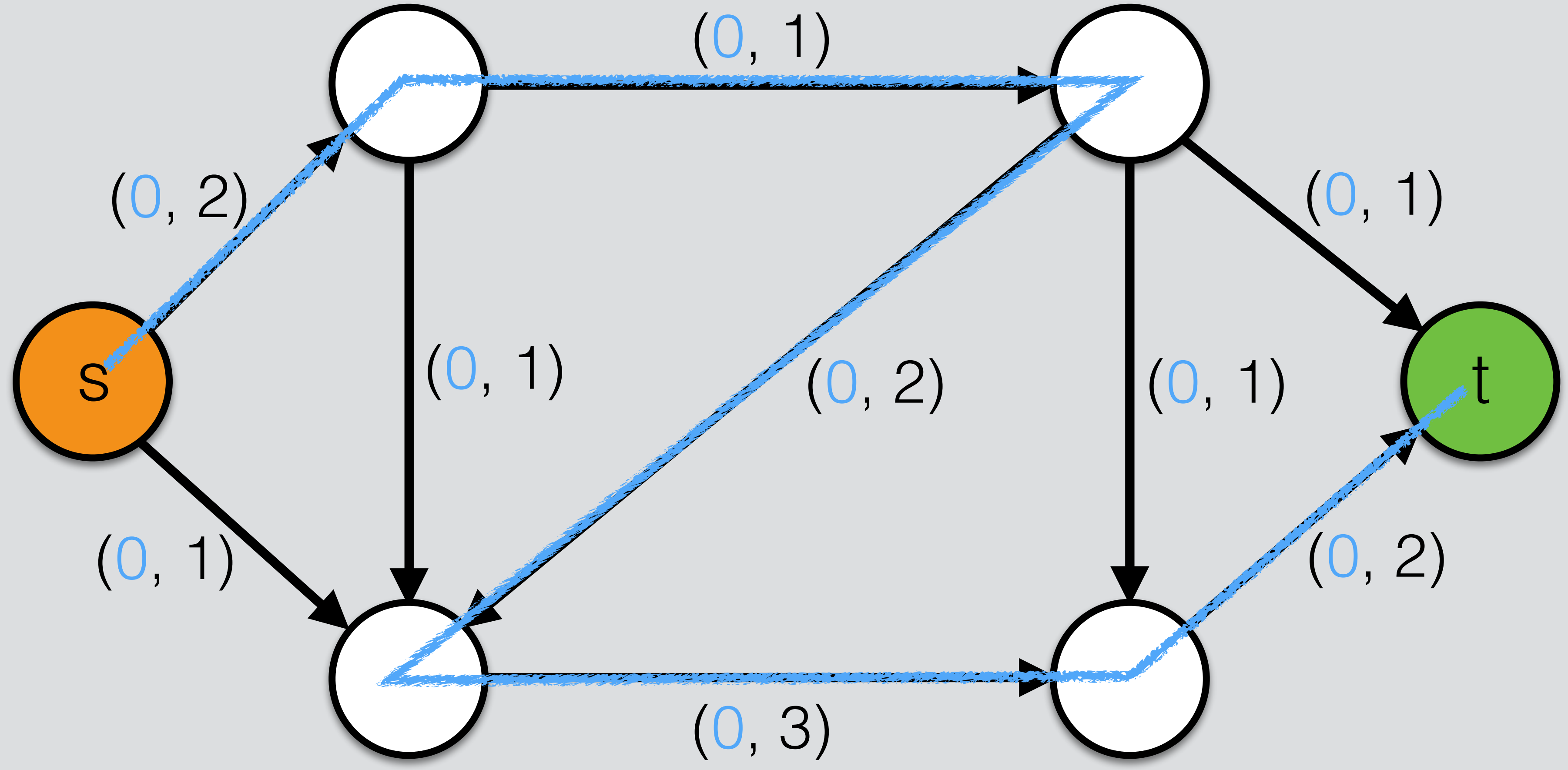
Input graph:



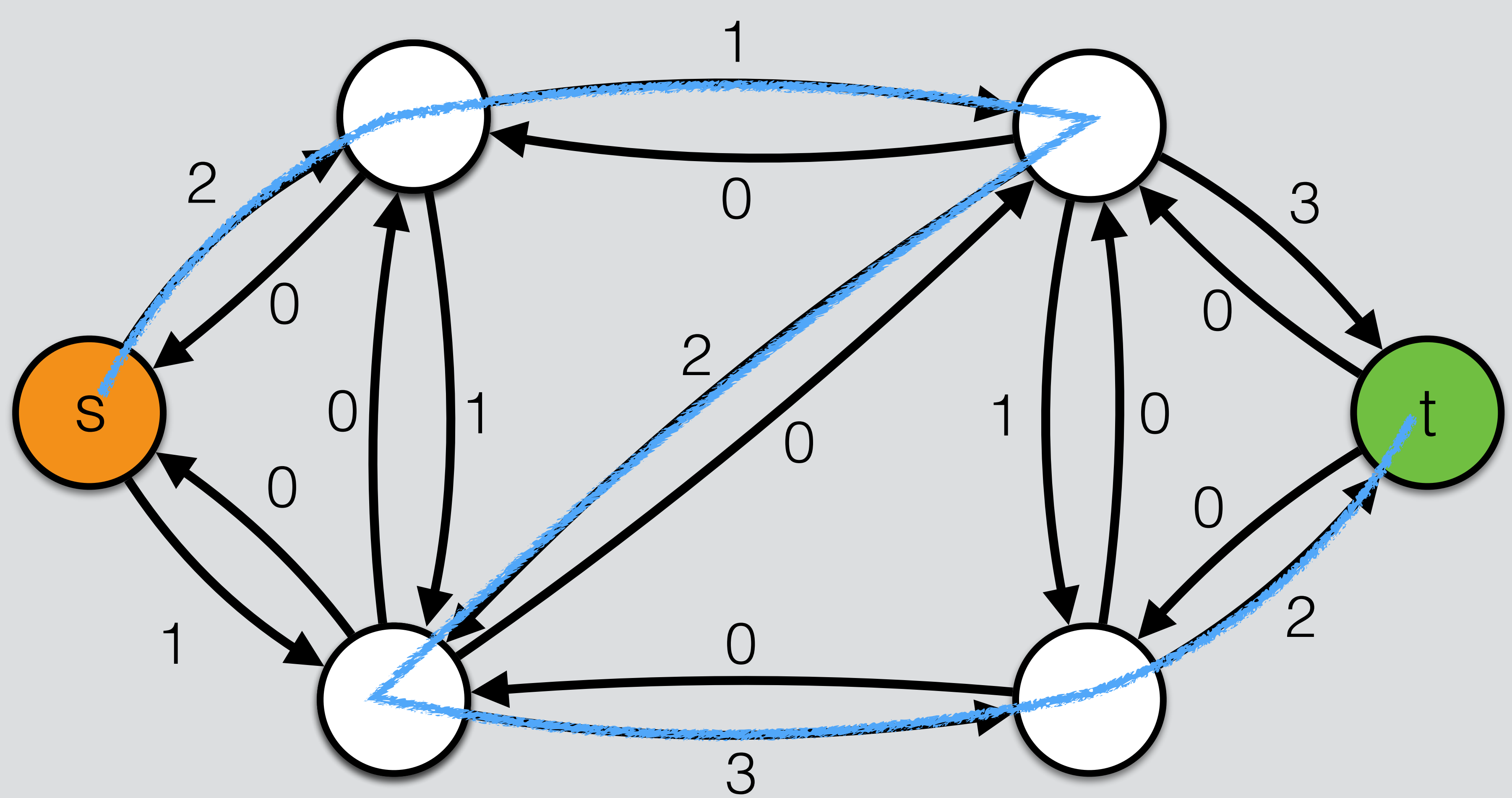
Residual graph:



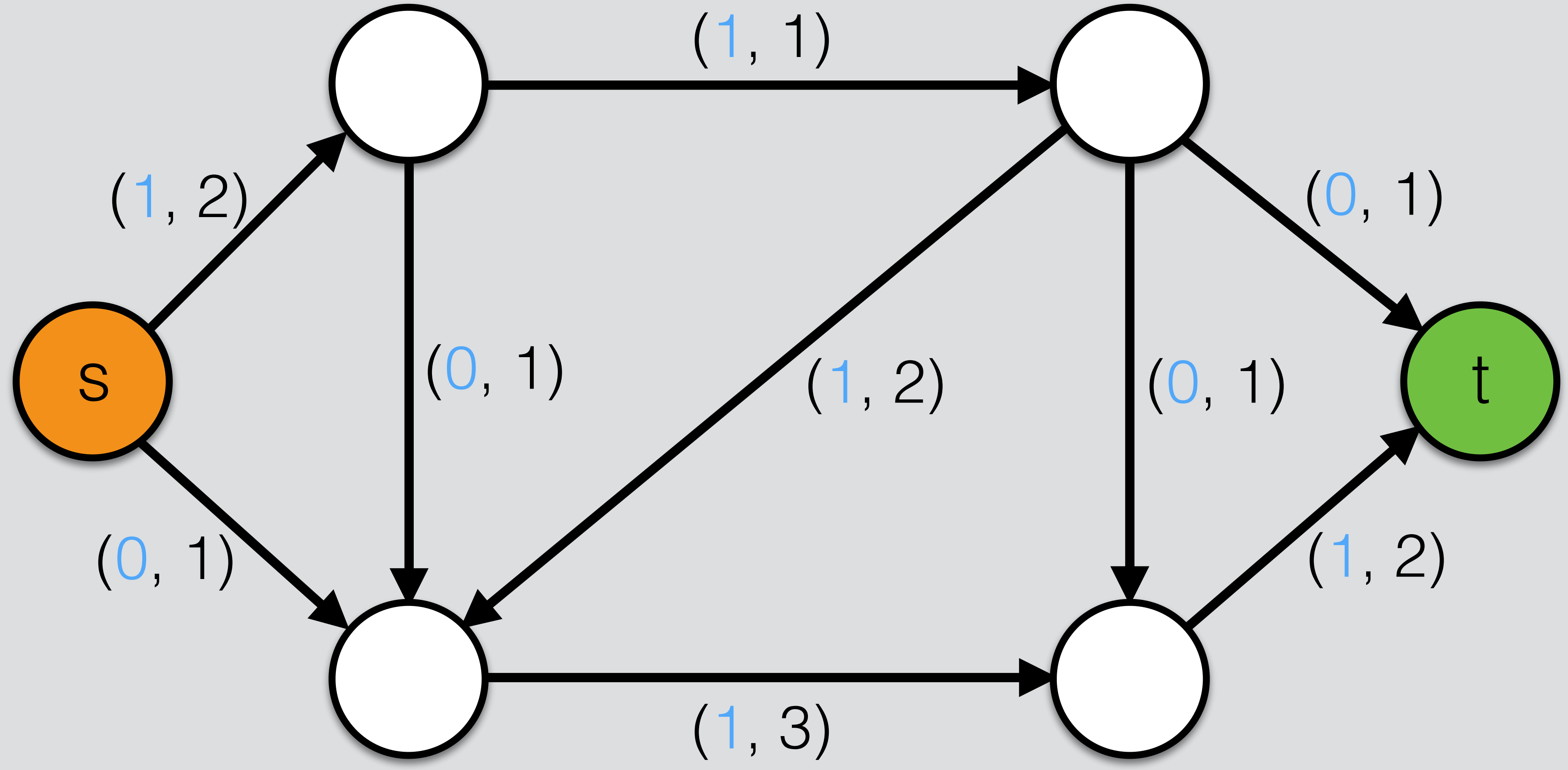
Input graph:



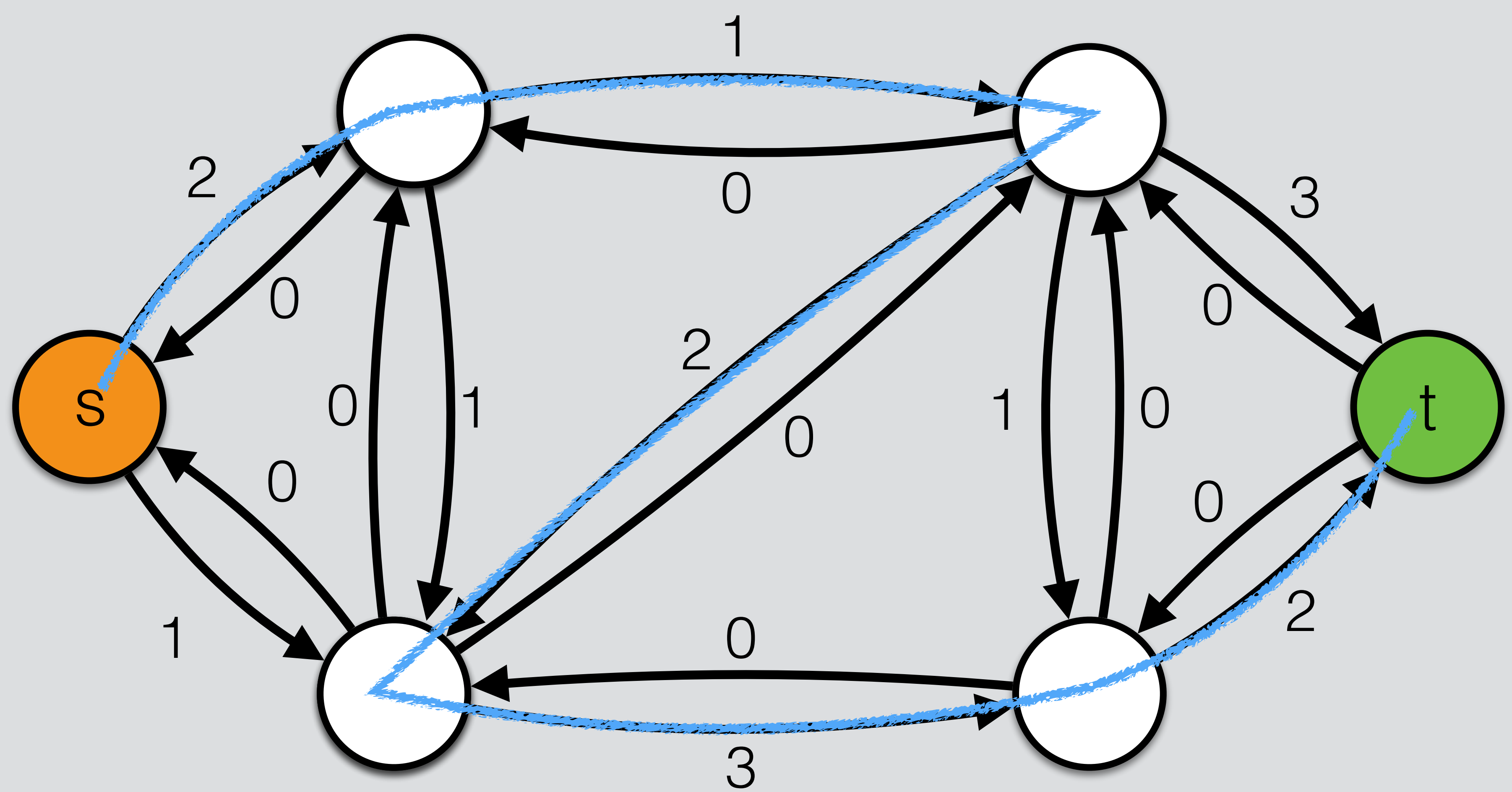
Residual graph:



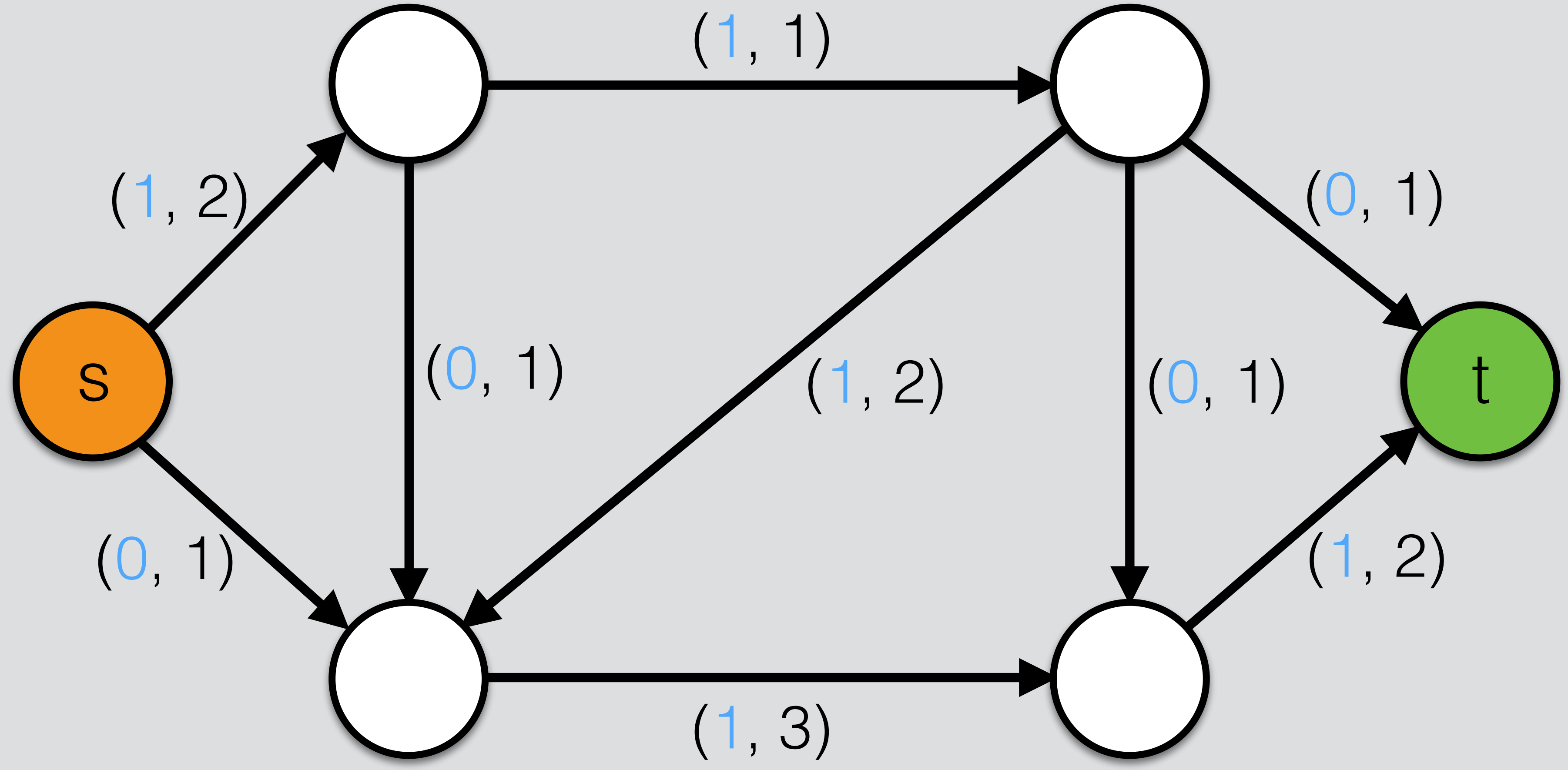
Input graph:



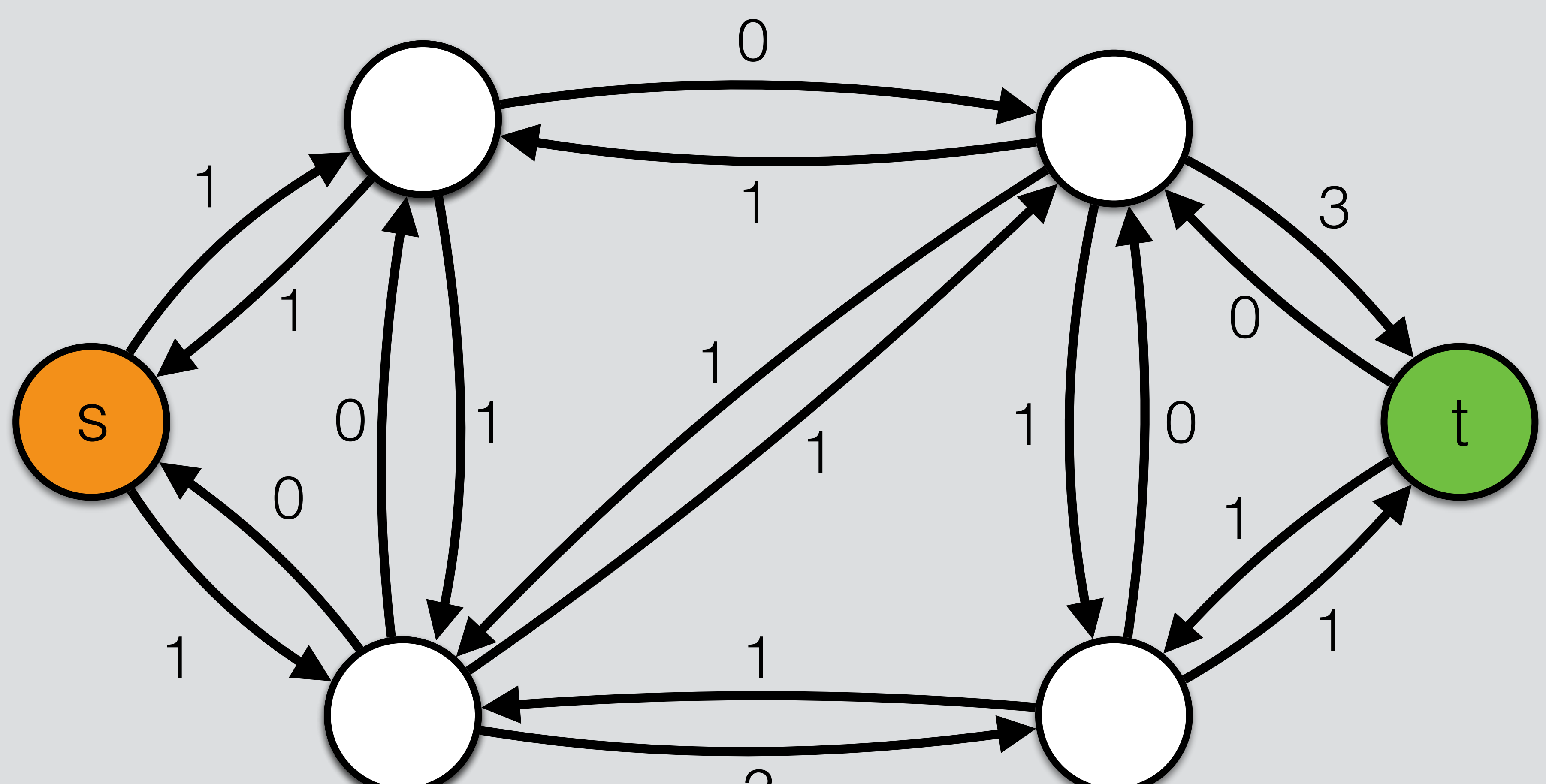
Residual graph:



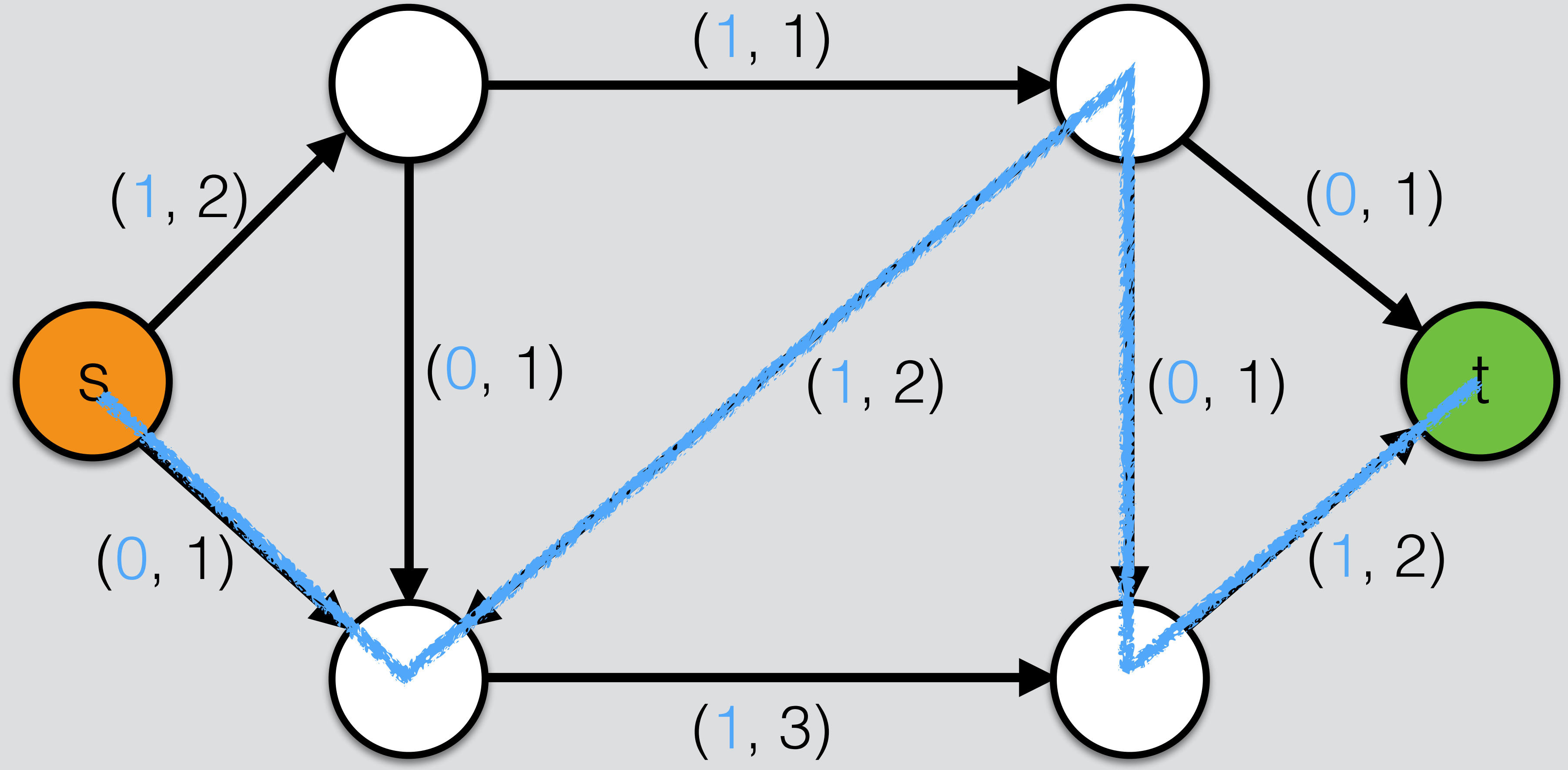
Input graph:



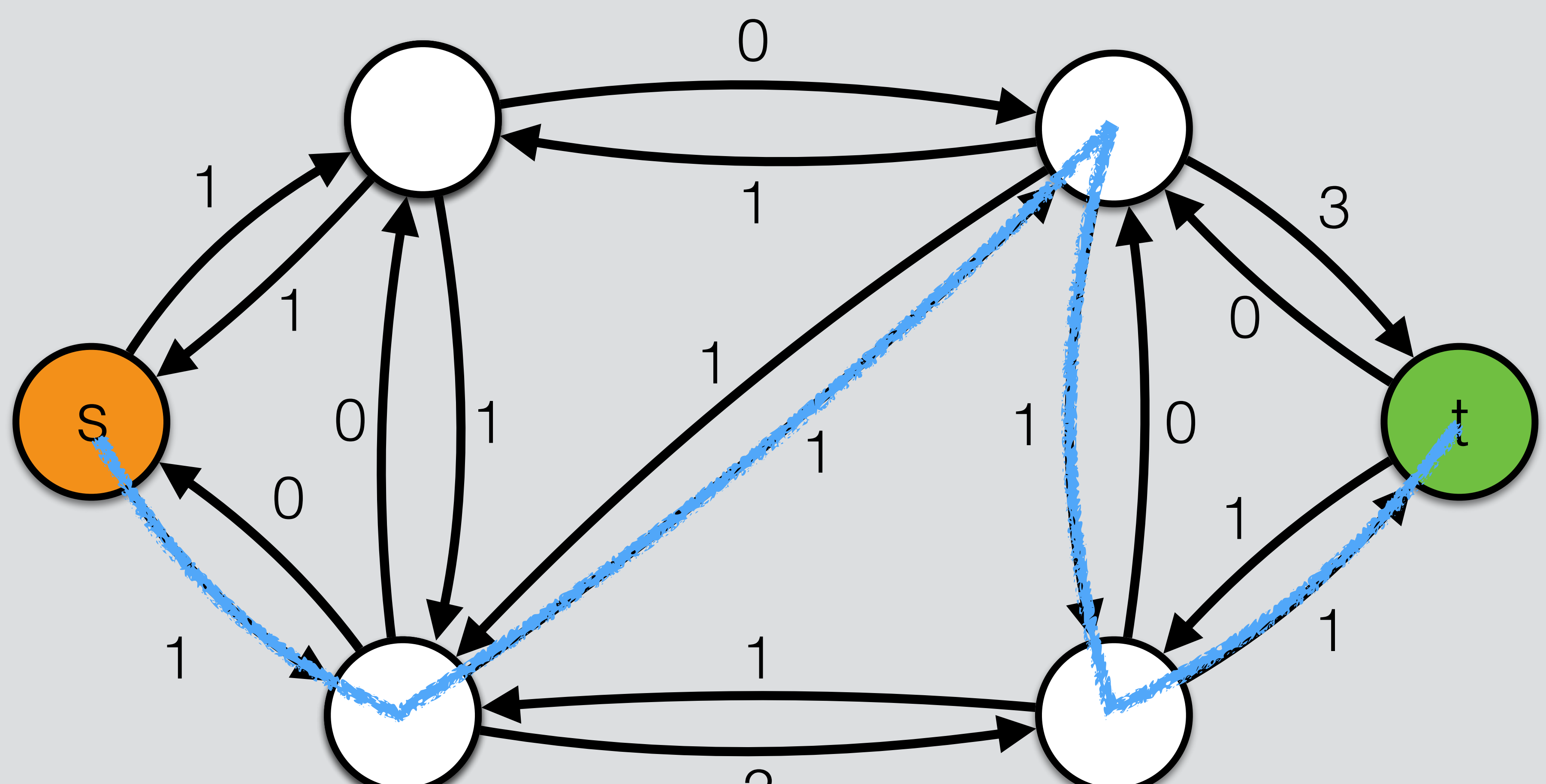
Residual graph:



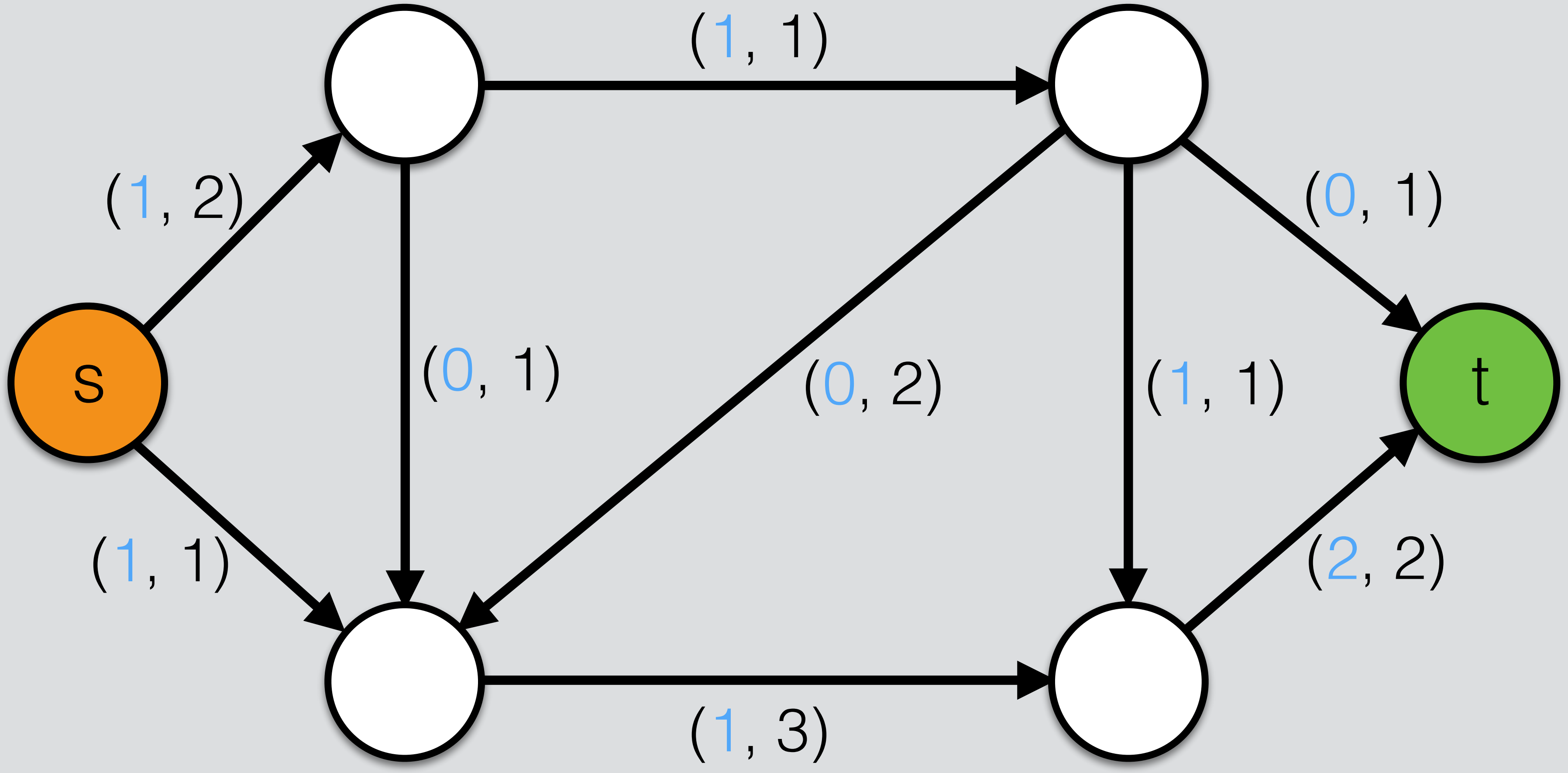
Input graph:



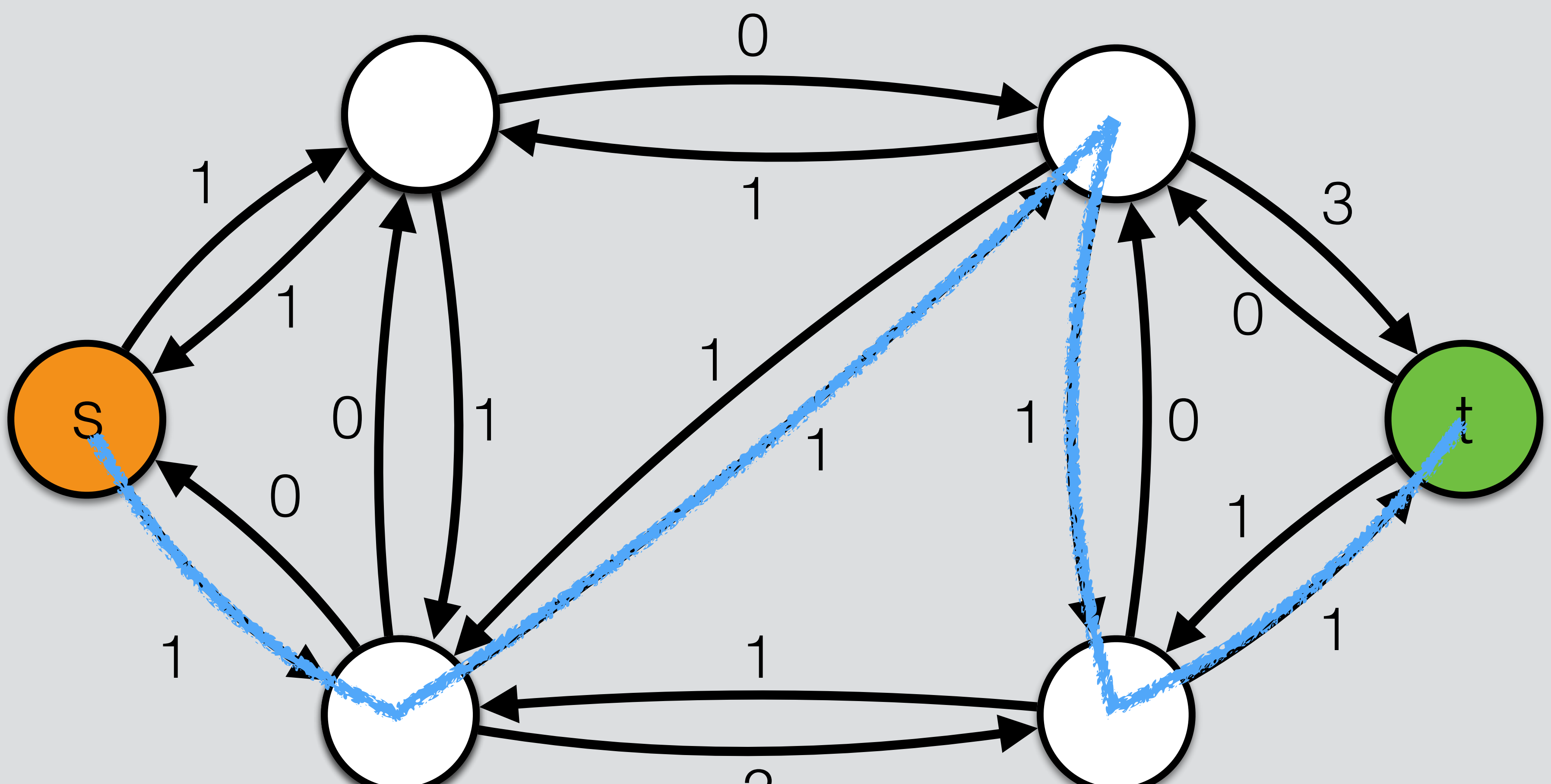
Residual graph:



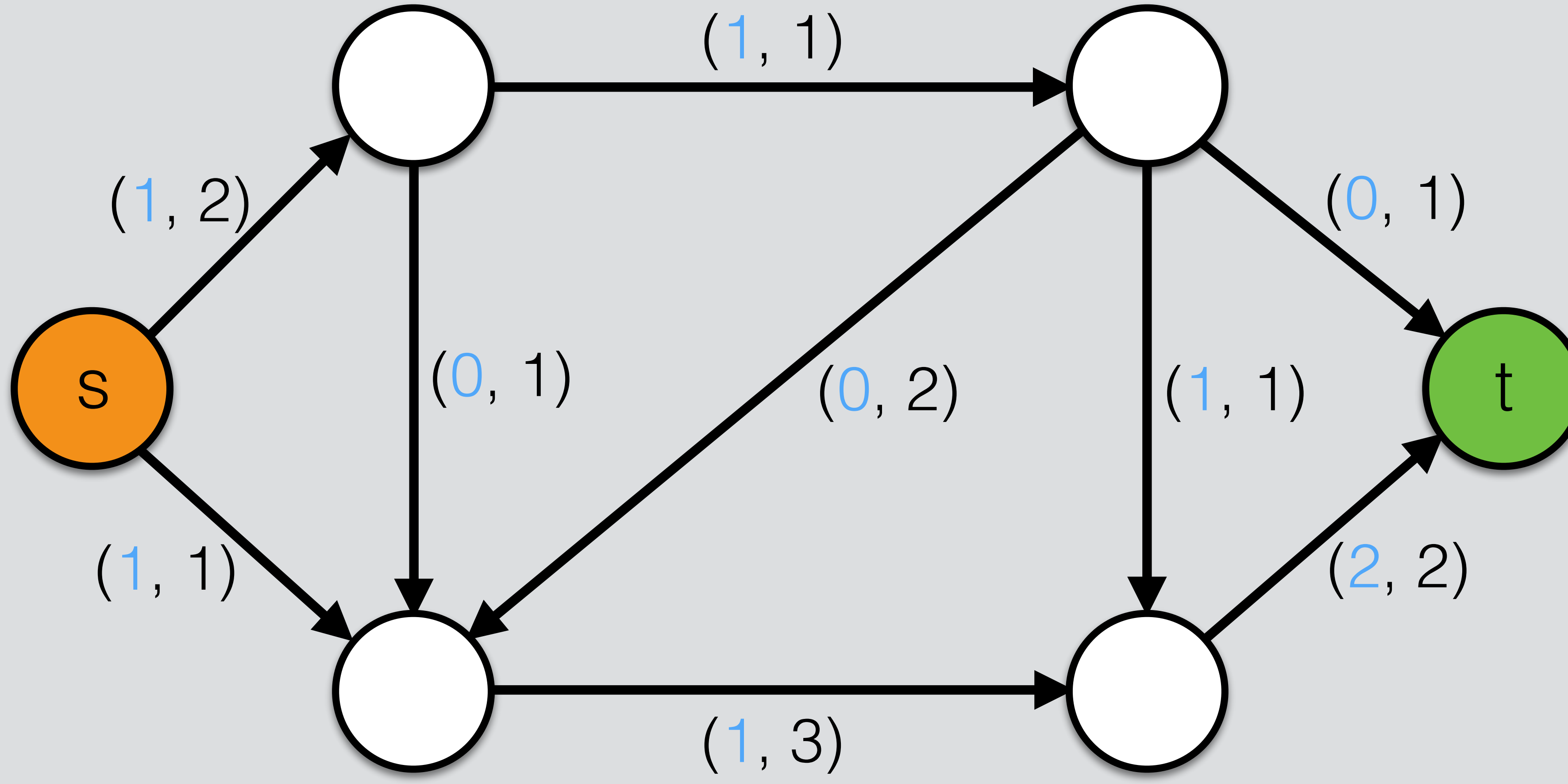
Input graph:



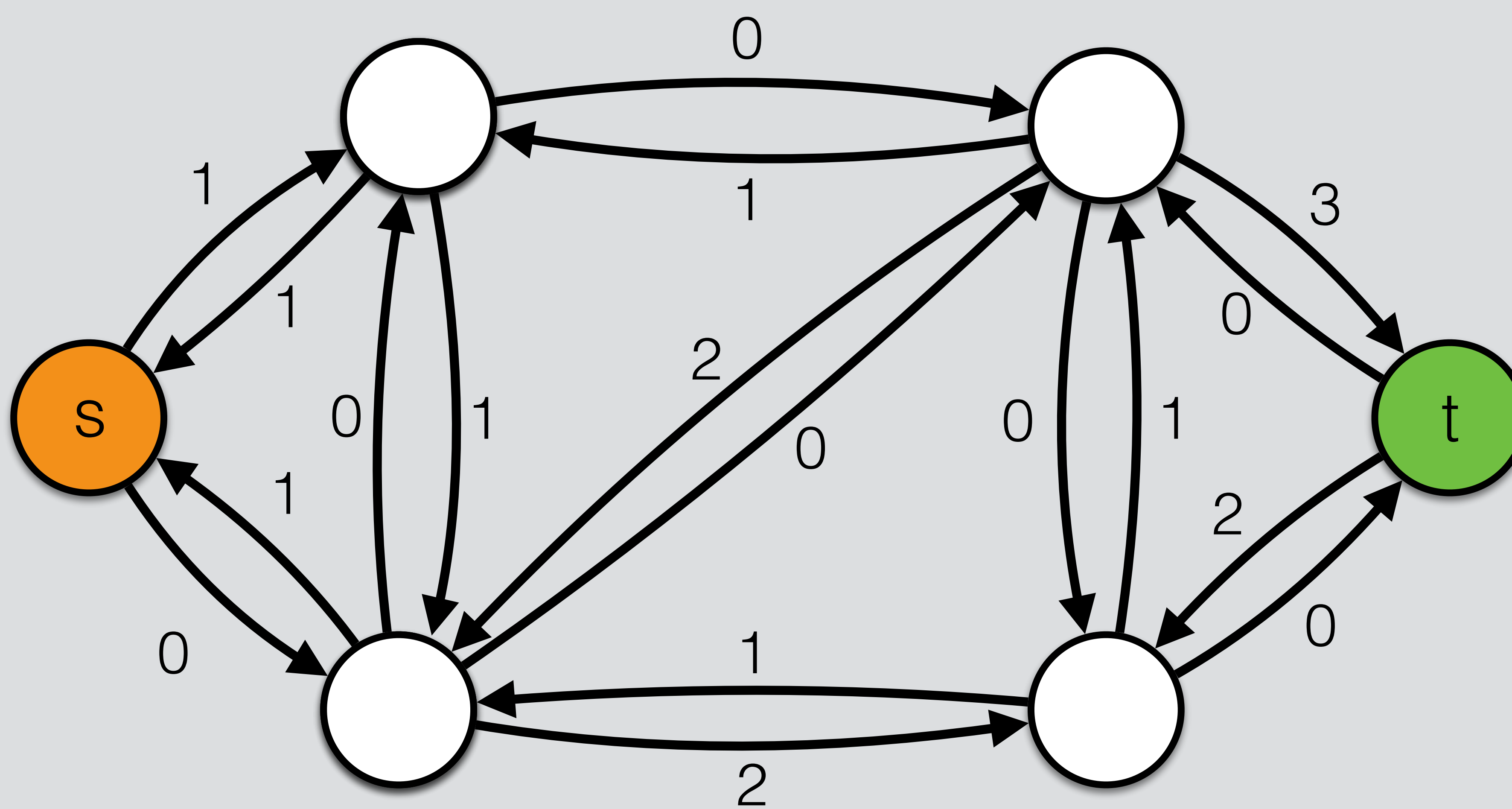
Residual graph:



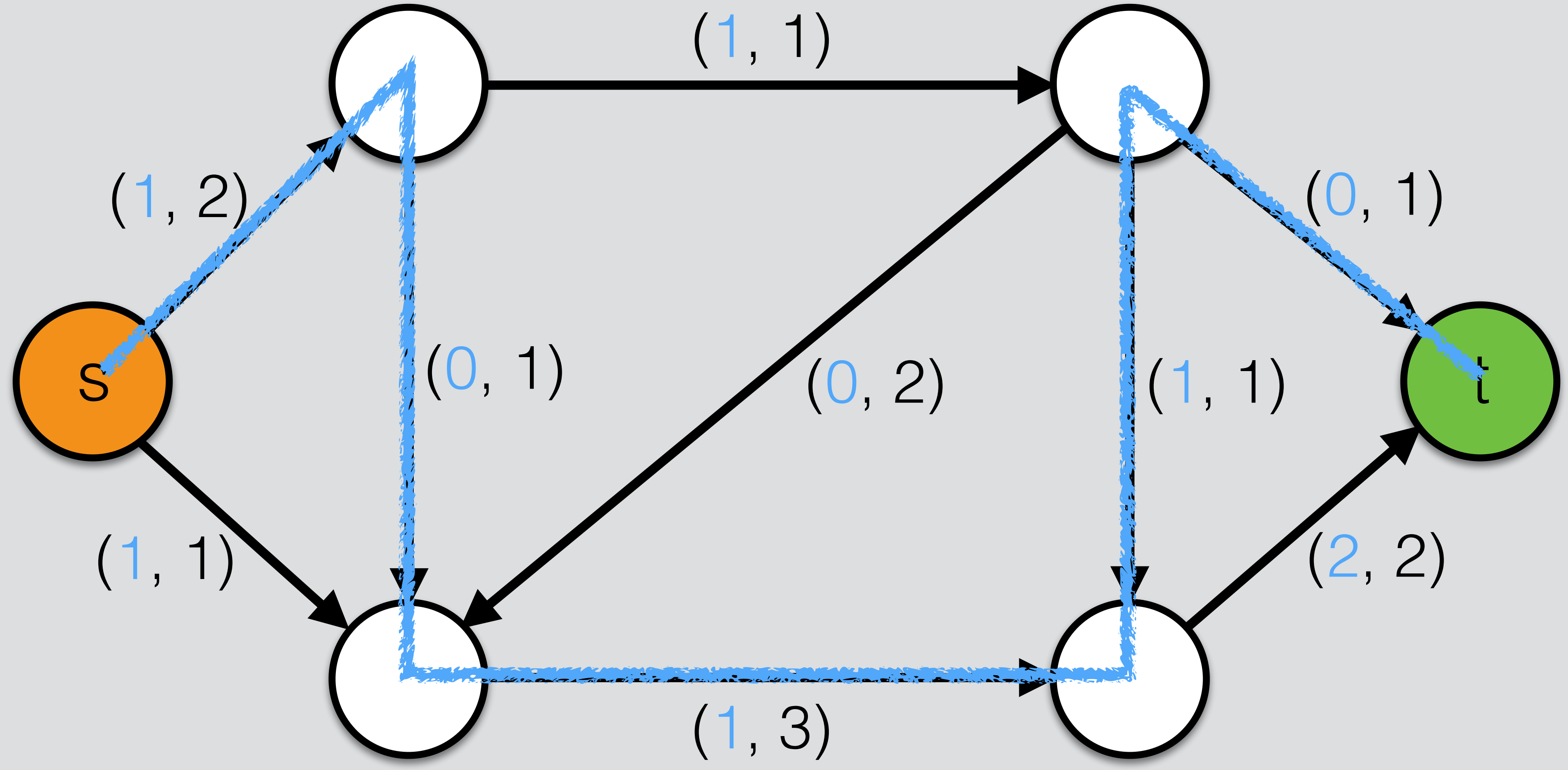
Input graph:



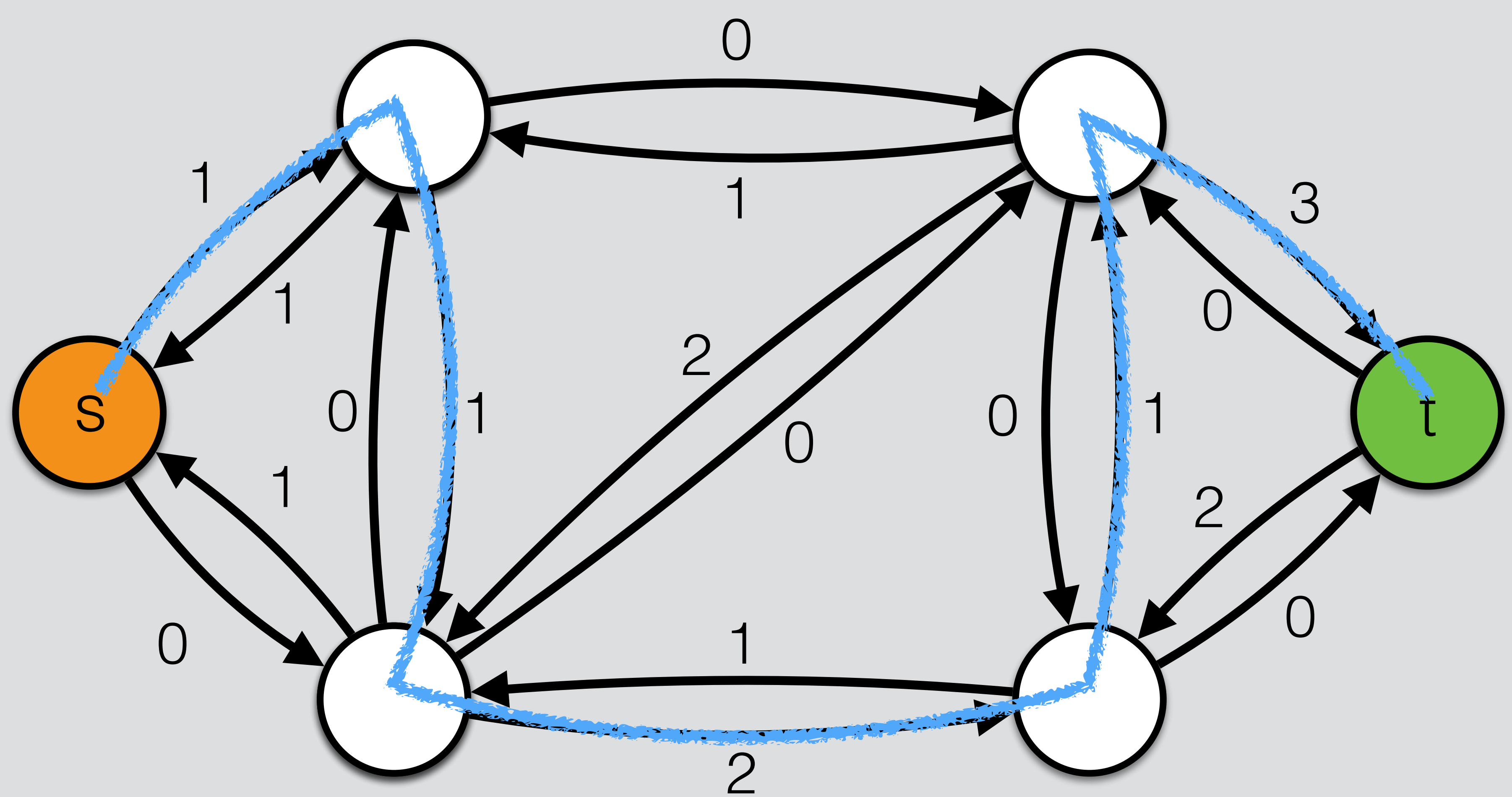
Residual graph:



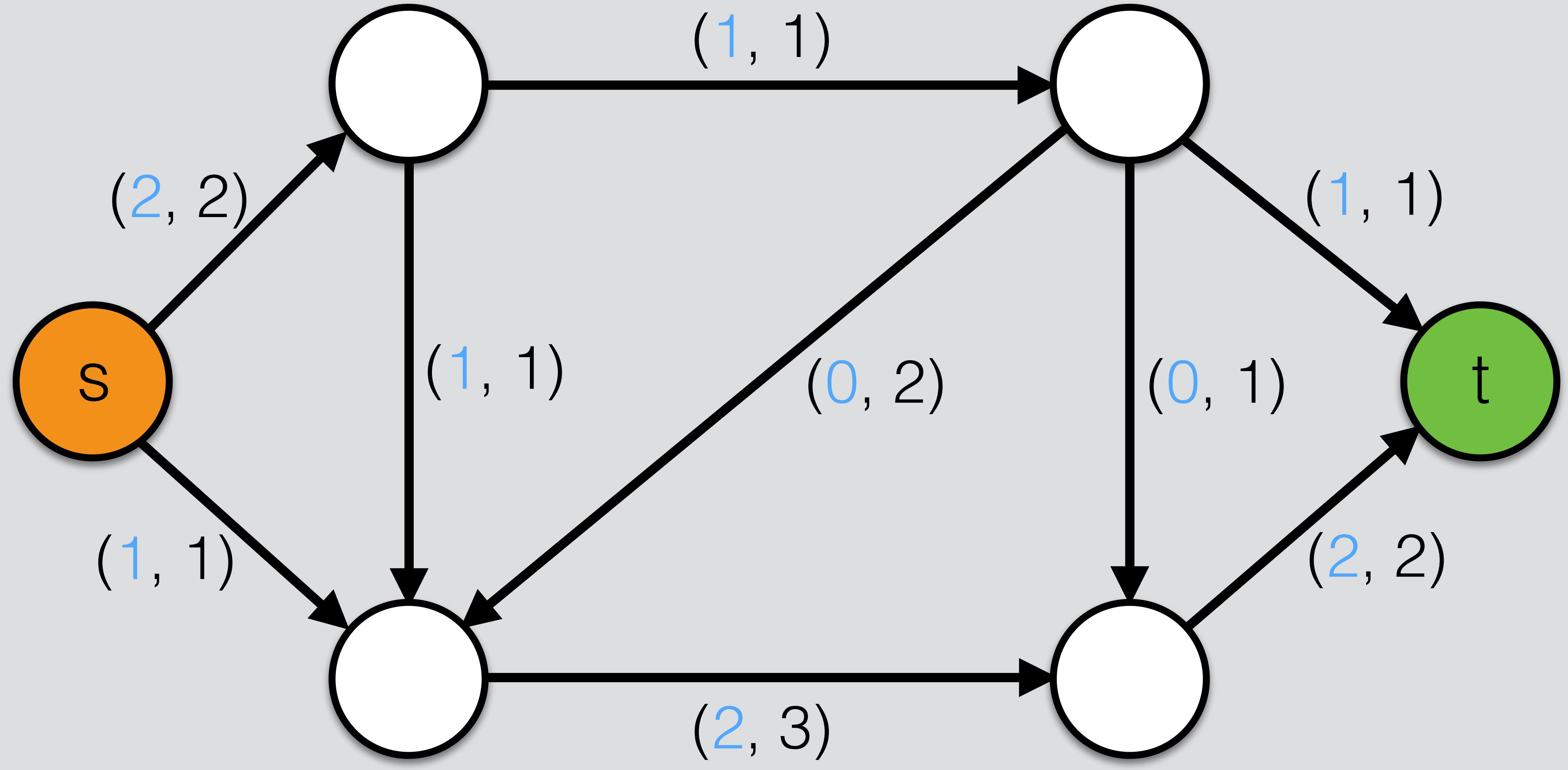
Input graph:



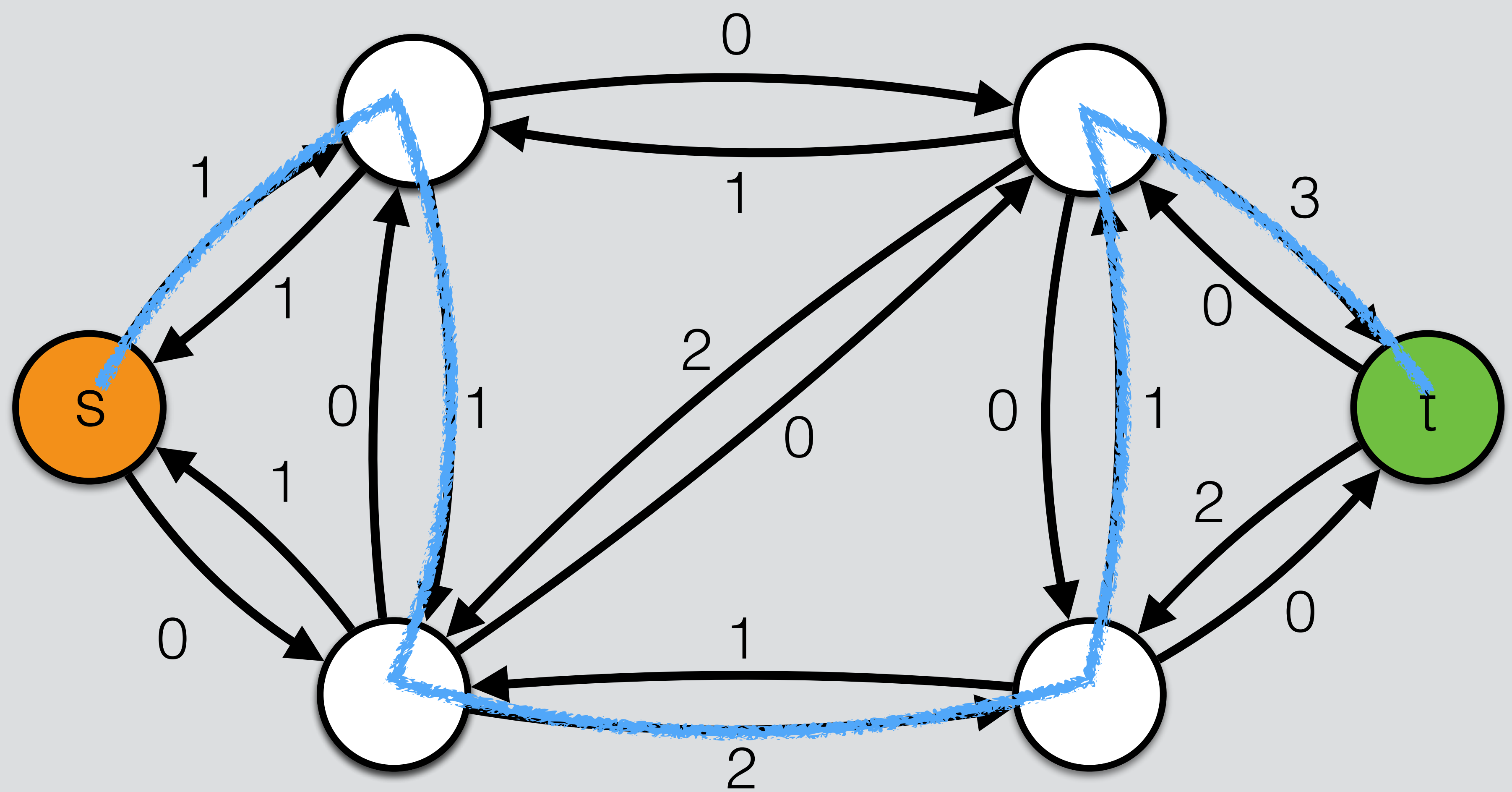
Residual graph:



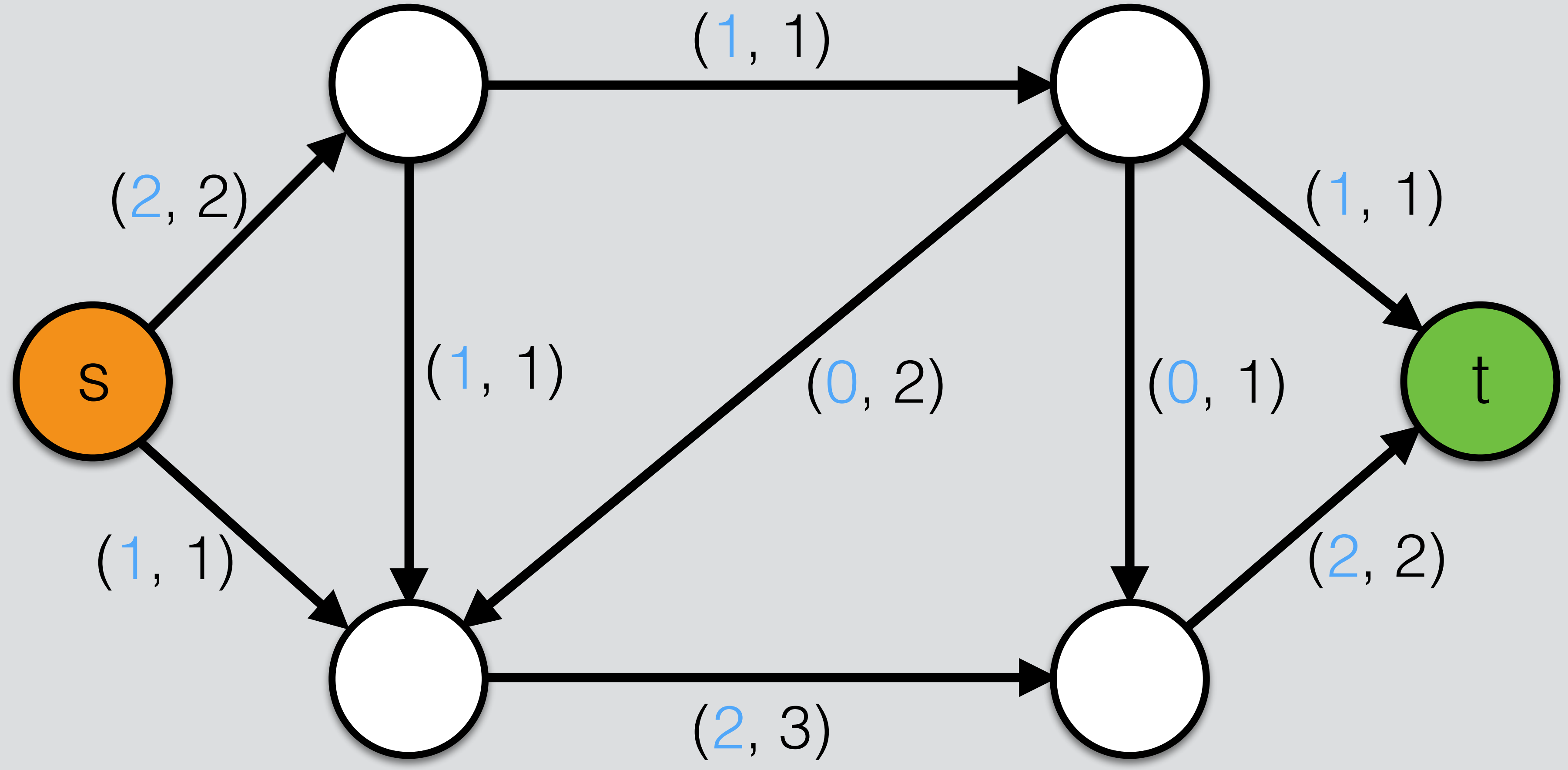
Input graph:



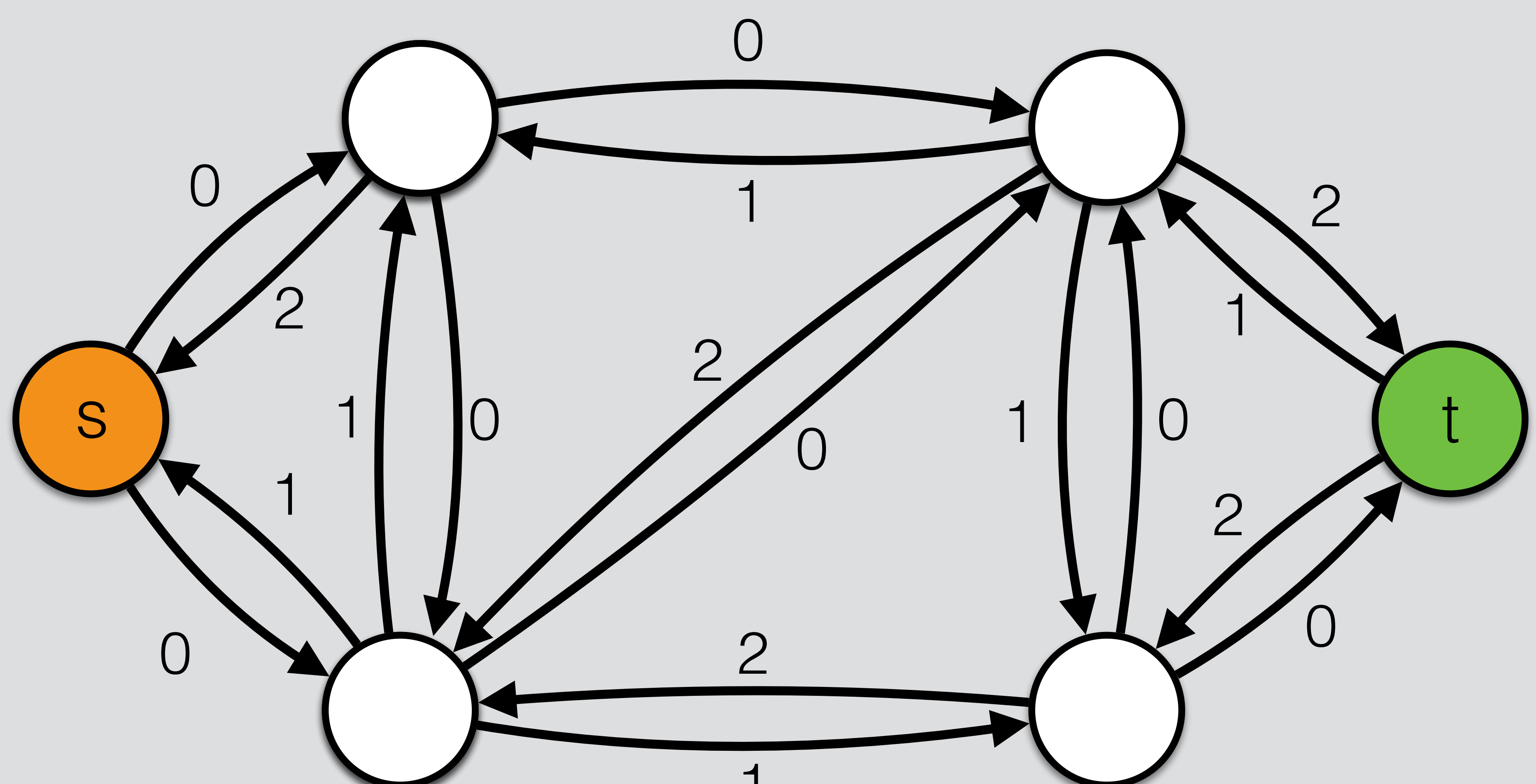
Residual graph:



Input graph:



Residual graph:



Ford-Fulkerson Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`)

Create an empty map `F`.

for (every edge (v_i, v_j) in E):

    if ((v_j, v_i) is not in `E`):

        Add (v_j, v_i) to `E` with capacity 0.

    $F[(v_i, v_j)] = 0;$

while (true):

    Let p = an augmenting path in residual of G.

    If no `p` exists, `break;`

    f_m = minimum weight of edge in `p`.

    Update flow along edges of `p` by f_m.

Setup

Increase flow via aug. paths

Ford-Fulkerson Running Time

Ford-Fulkerson Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`)

, and max flow f_{max}

Create an empty map `F`. $\Theta(n)$

for (every edge (v_i, v_j) in E): $\Theta(m)$

    if ((v_j, v_i) is not in `E`):

        Add (v_j, v_i) to `E` with capacity 0.

    `F[(vi, vj)] = 0;`

while (true):

    Let p = an augmenting path in residual of G. $\Theta(n+m)$

    If no `p` exists, `break;`

    f_{aug} = minimum weight of edge in `p`. $\Theta(n)$

    Update flow along edges of `p` by f_{aug}.

$O(f_{max}*(n+m))$ total

Ford-Fulkerson Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`), and max flow f_{max}

Create an empty map `F`.

for (every edge `(vᵢ, vⱼ)` in E):

    if (`(vⱼ, vᵢ)` is not in `E`):

        Add (v_j, v_i) to `E` with capacity 0.

    `F[(vᵢ, vⱼ)] = 0;`

while (true):

    Let p = an augmenting path in residual of G.

    If no `p` exists, `break;`

    f_{aug} = minimum weight of edge in `p`.

    Update flow along edges of `p` by f_{aug}.

Θ(n)

Θ(m)

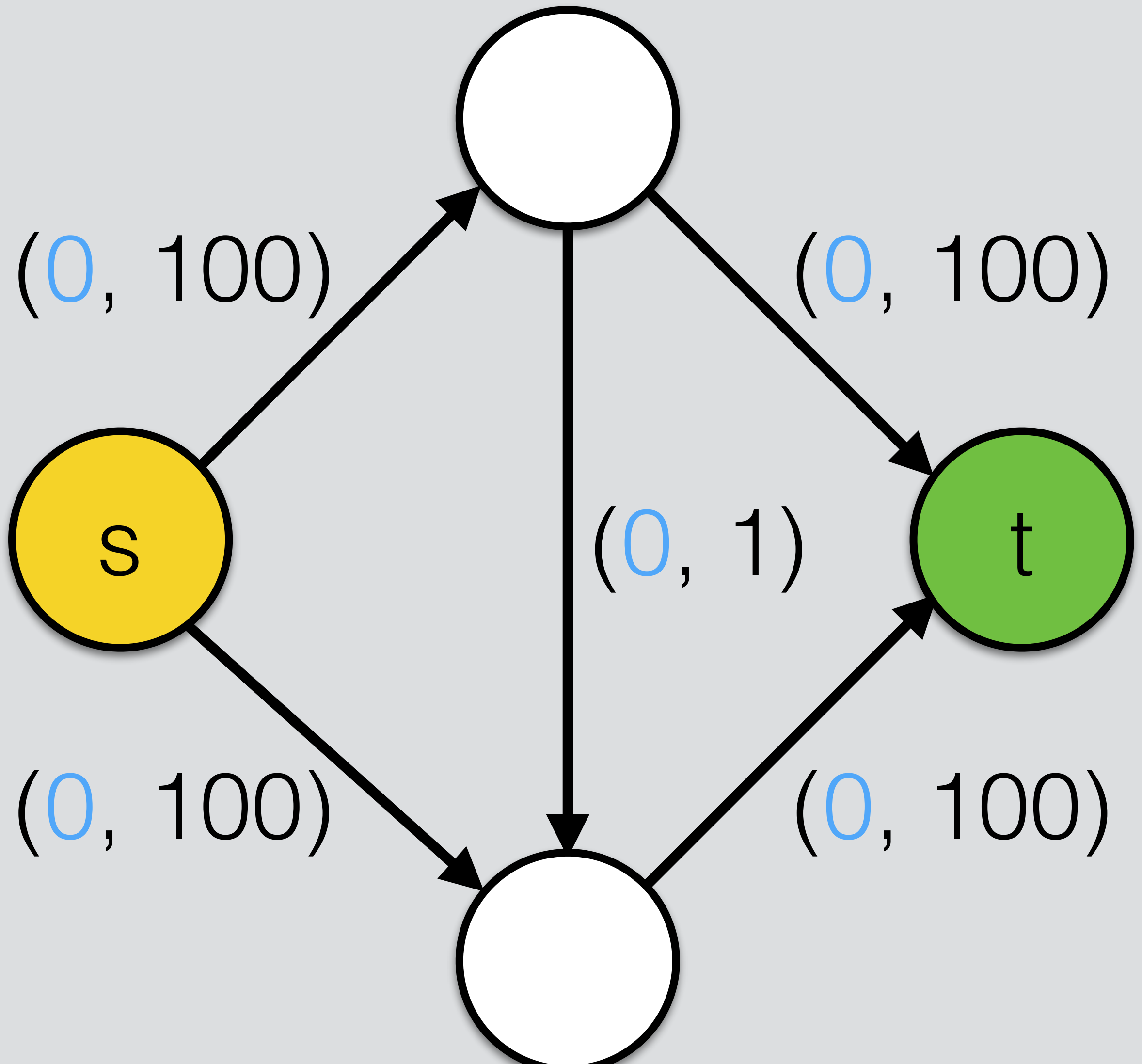
Θ(n+m)

Θ(n)

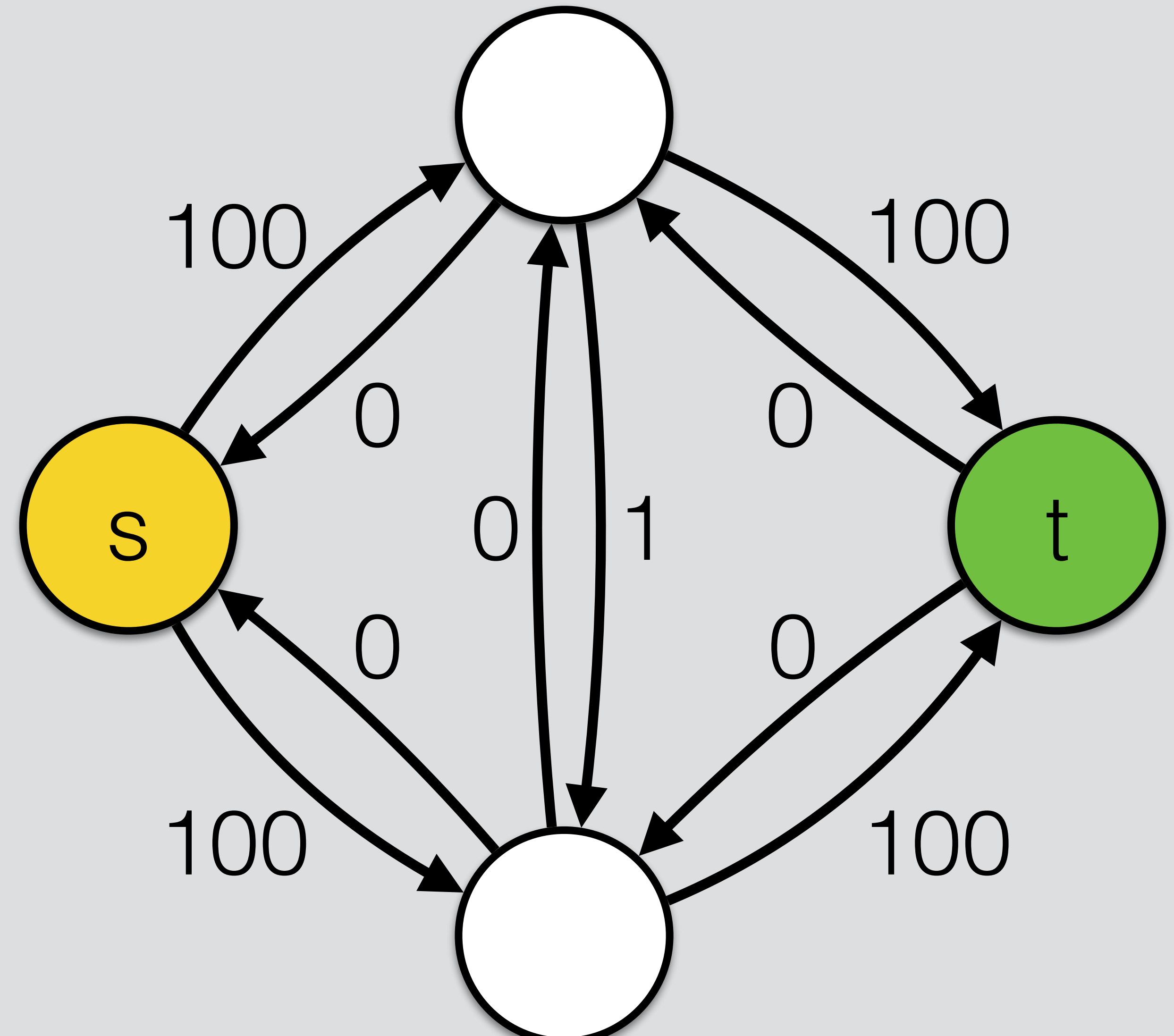
$O(f_{max}*(n+m))$ total

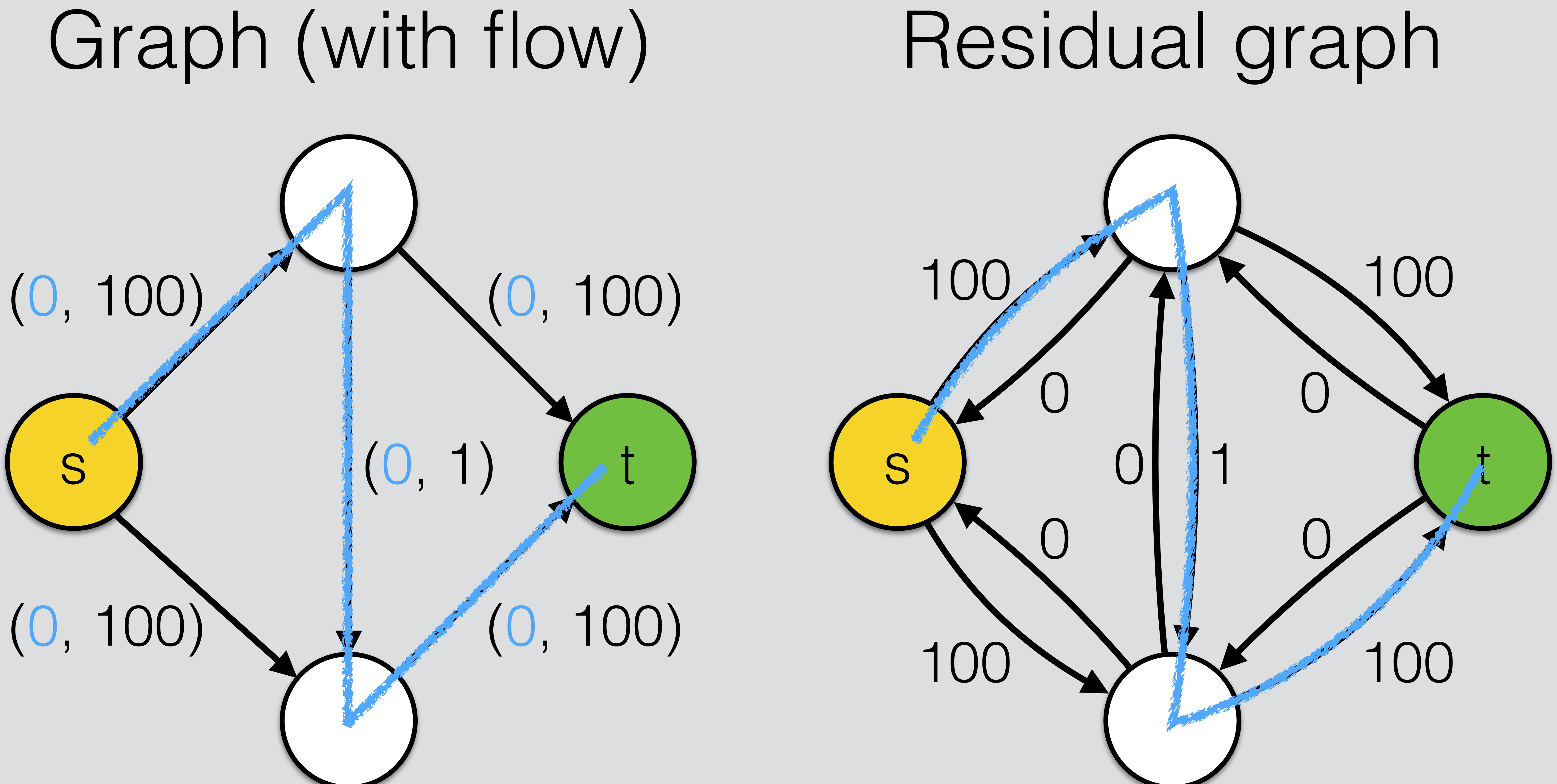
$O(f_{max}*(n + m))$ time

Graph (with flow)

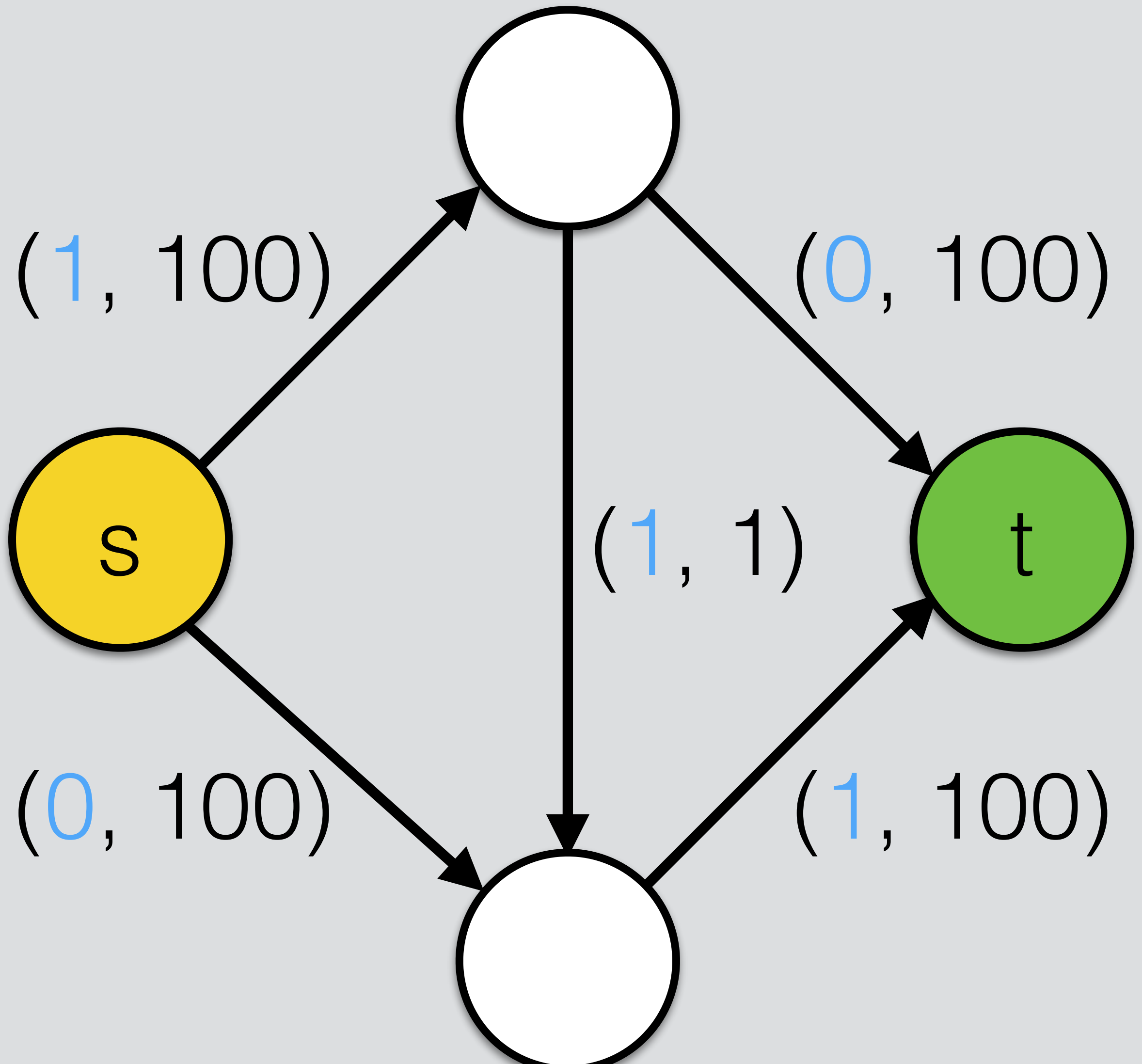


Residual graph

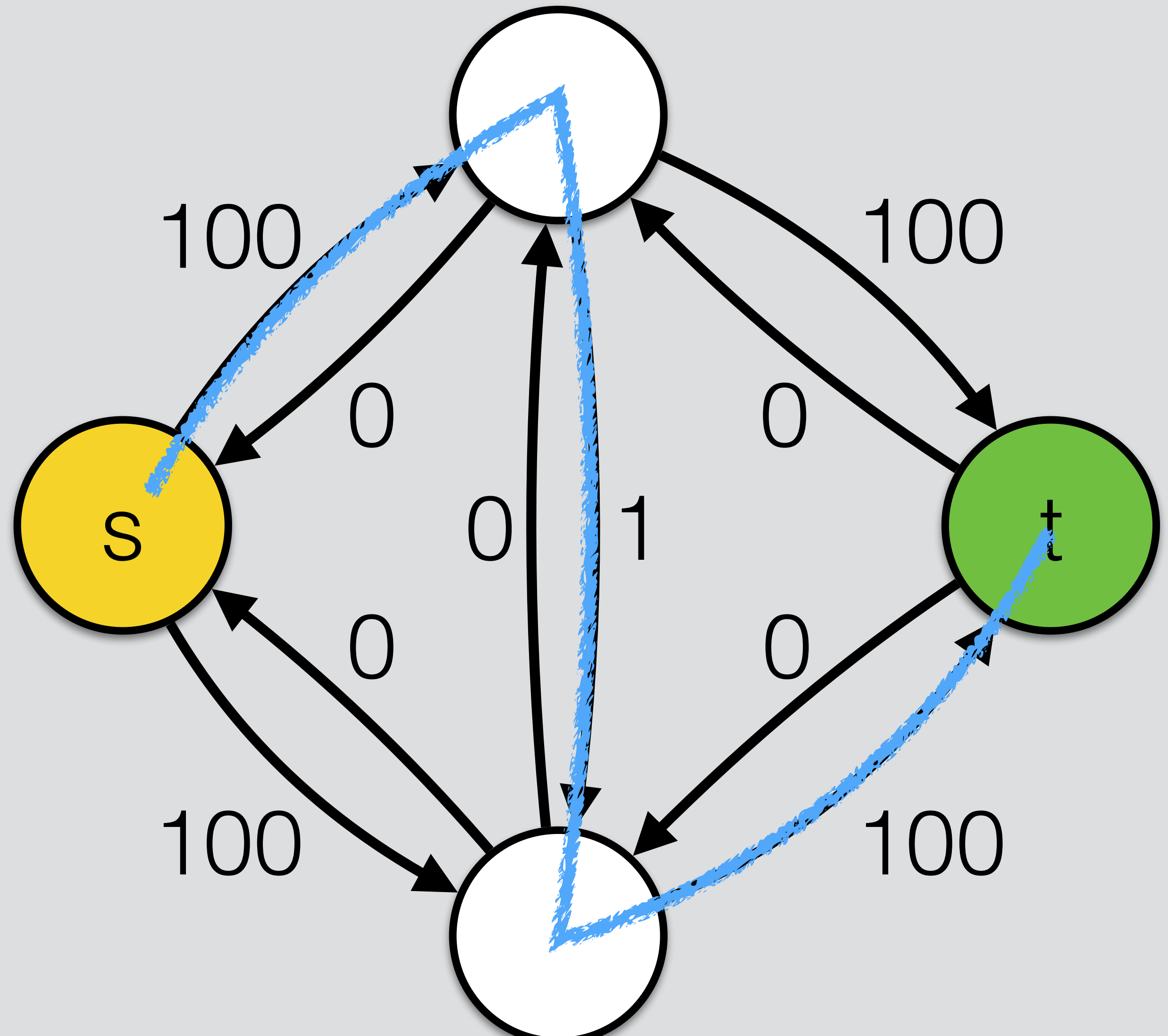



Graph (with flow)
(0, 100)
(0, 100)
(0, 1)
(0, 100)
(0, 100)
s
t
Residual graph
100
100
0
0
0
1
0
0
100
100
s
t

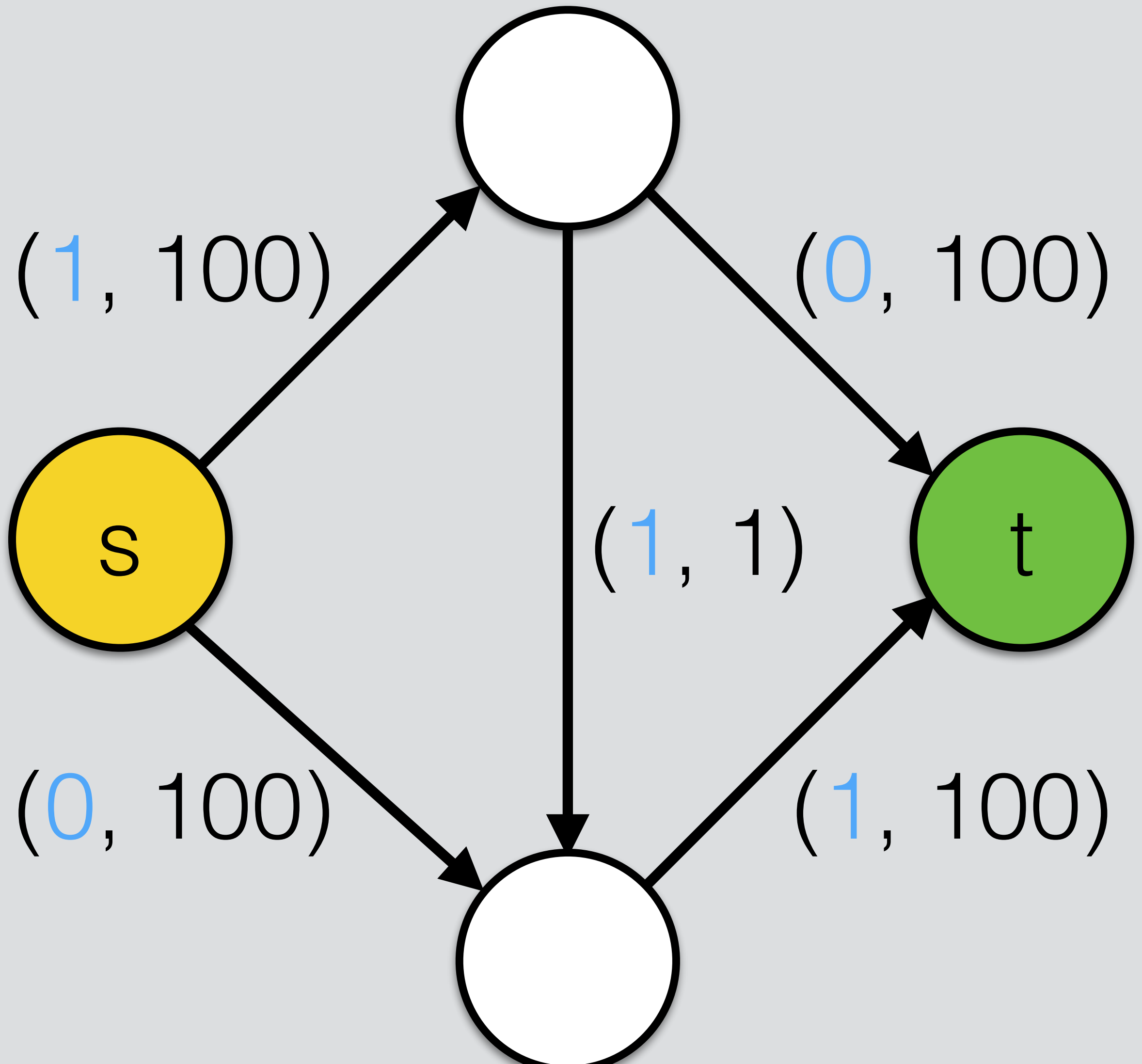
Graph (with flow)



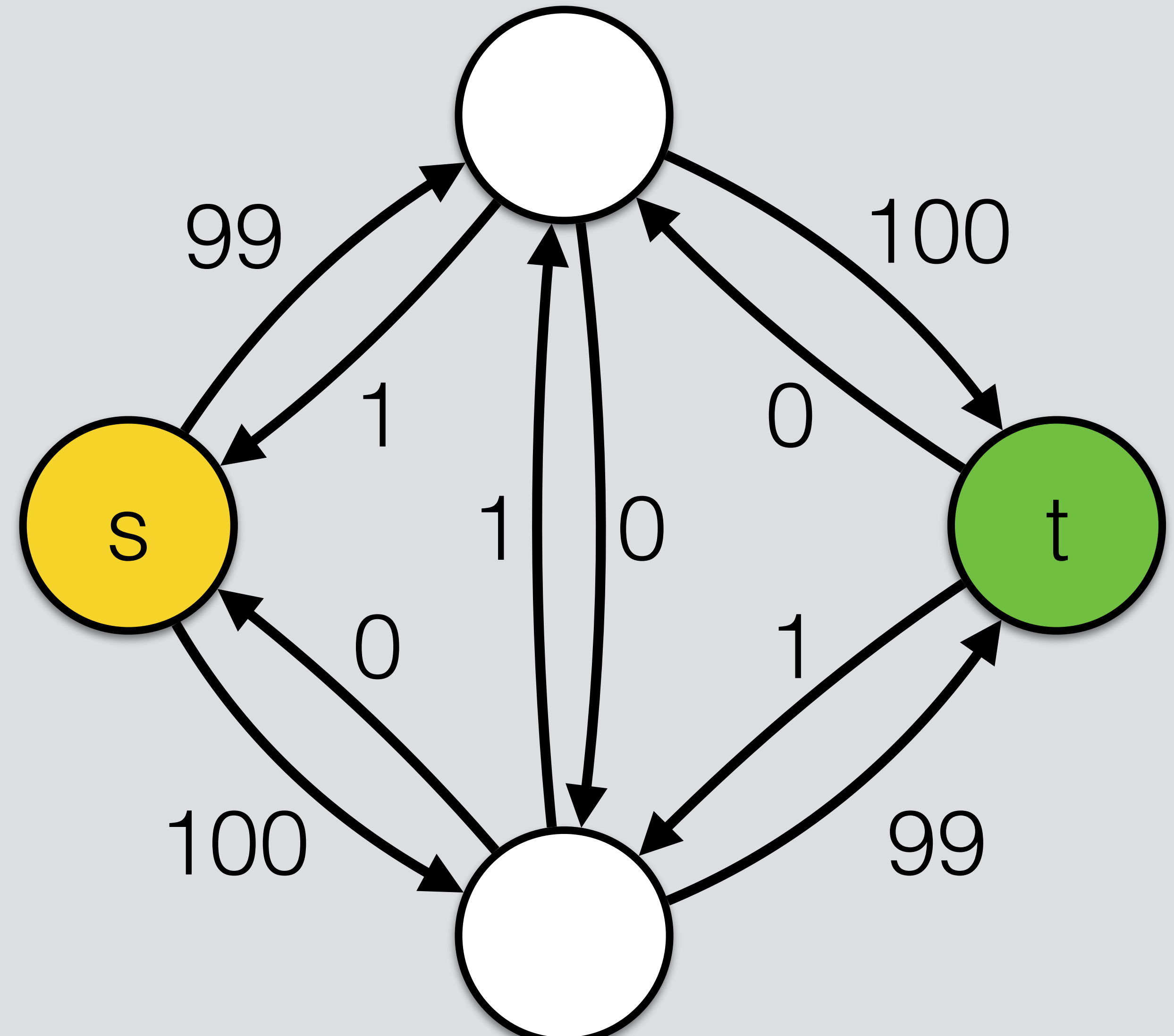
Residual graph



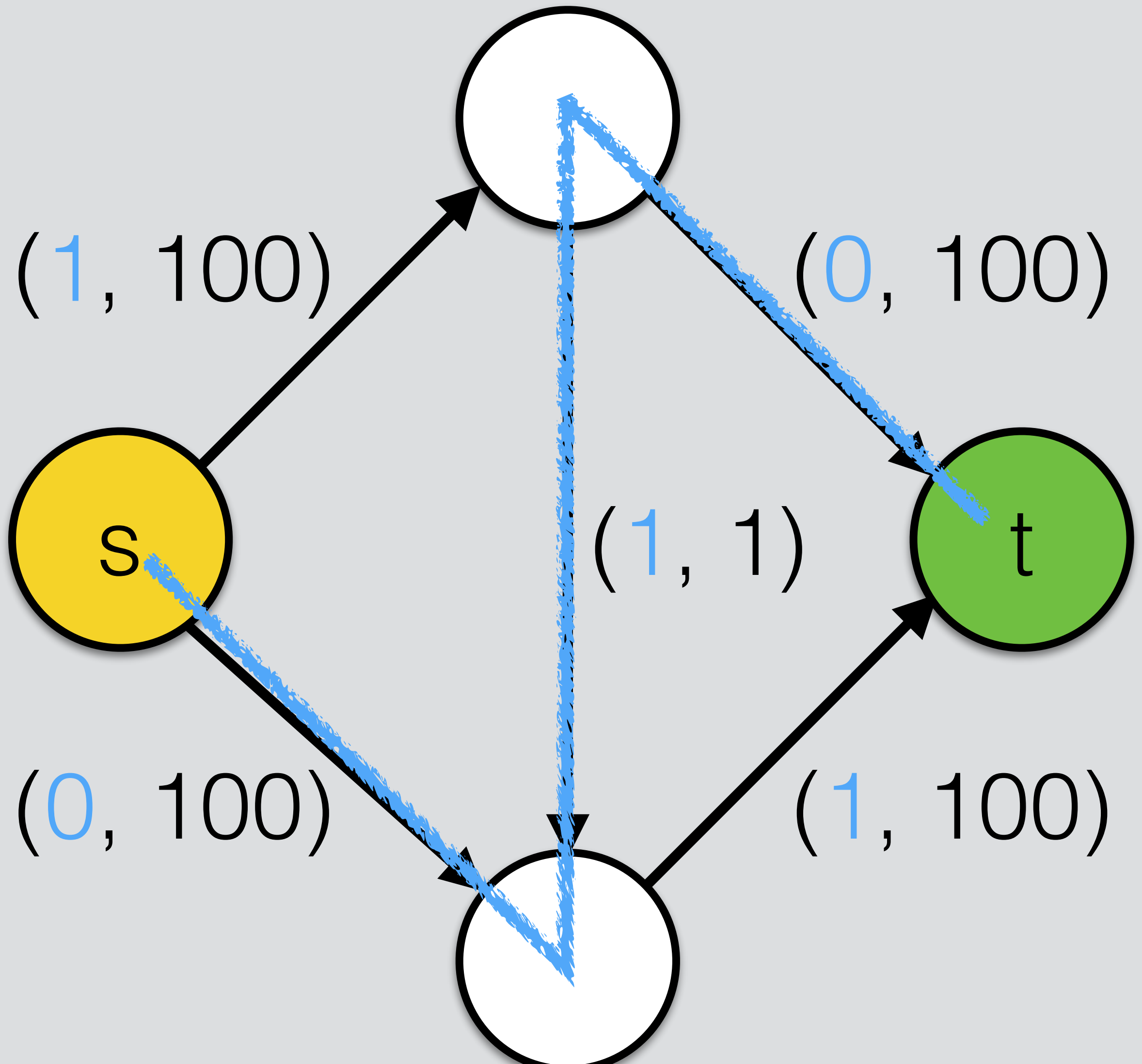
Graph (with flow)



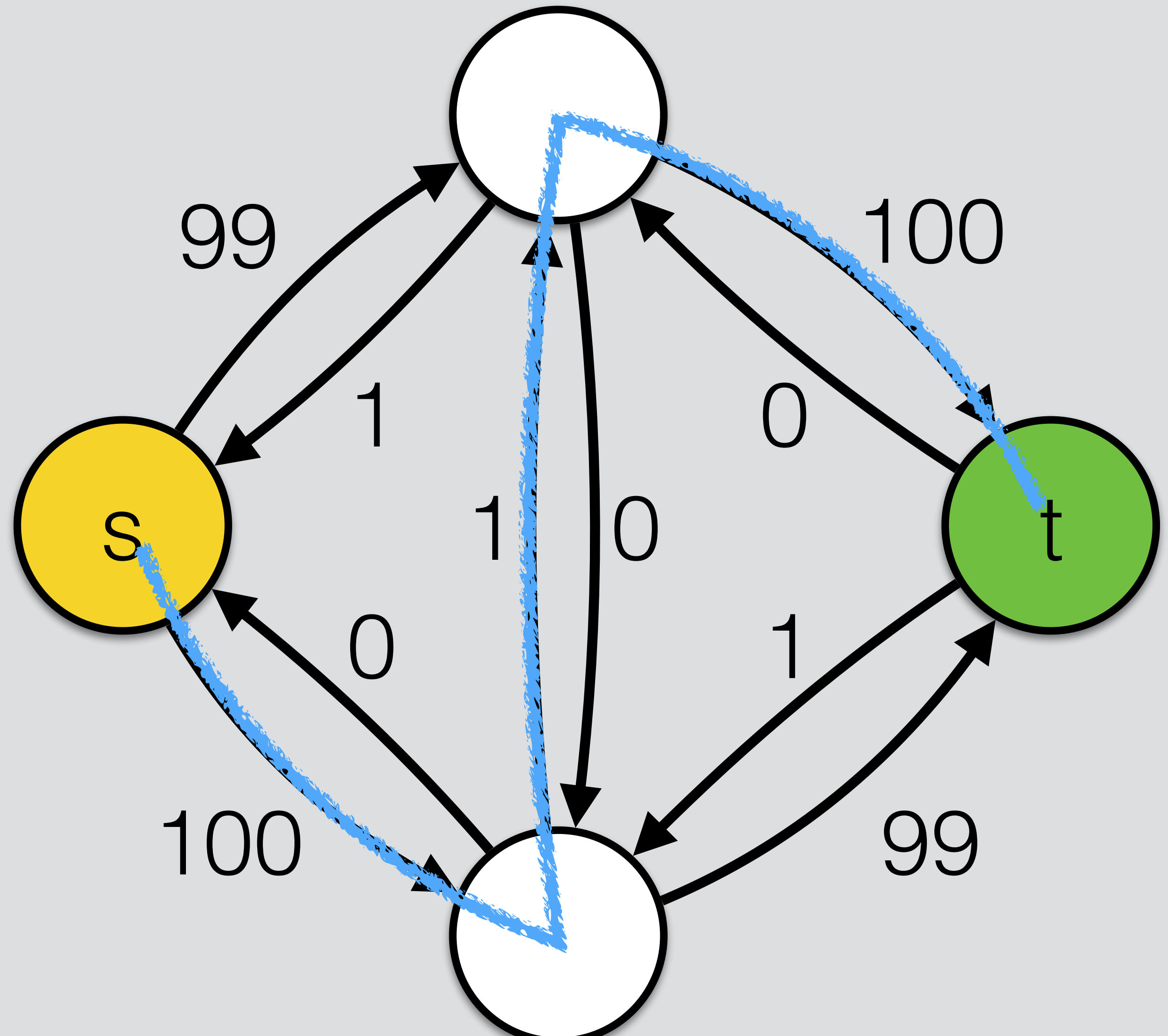
Residual graph



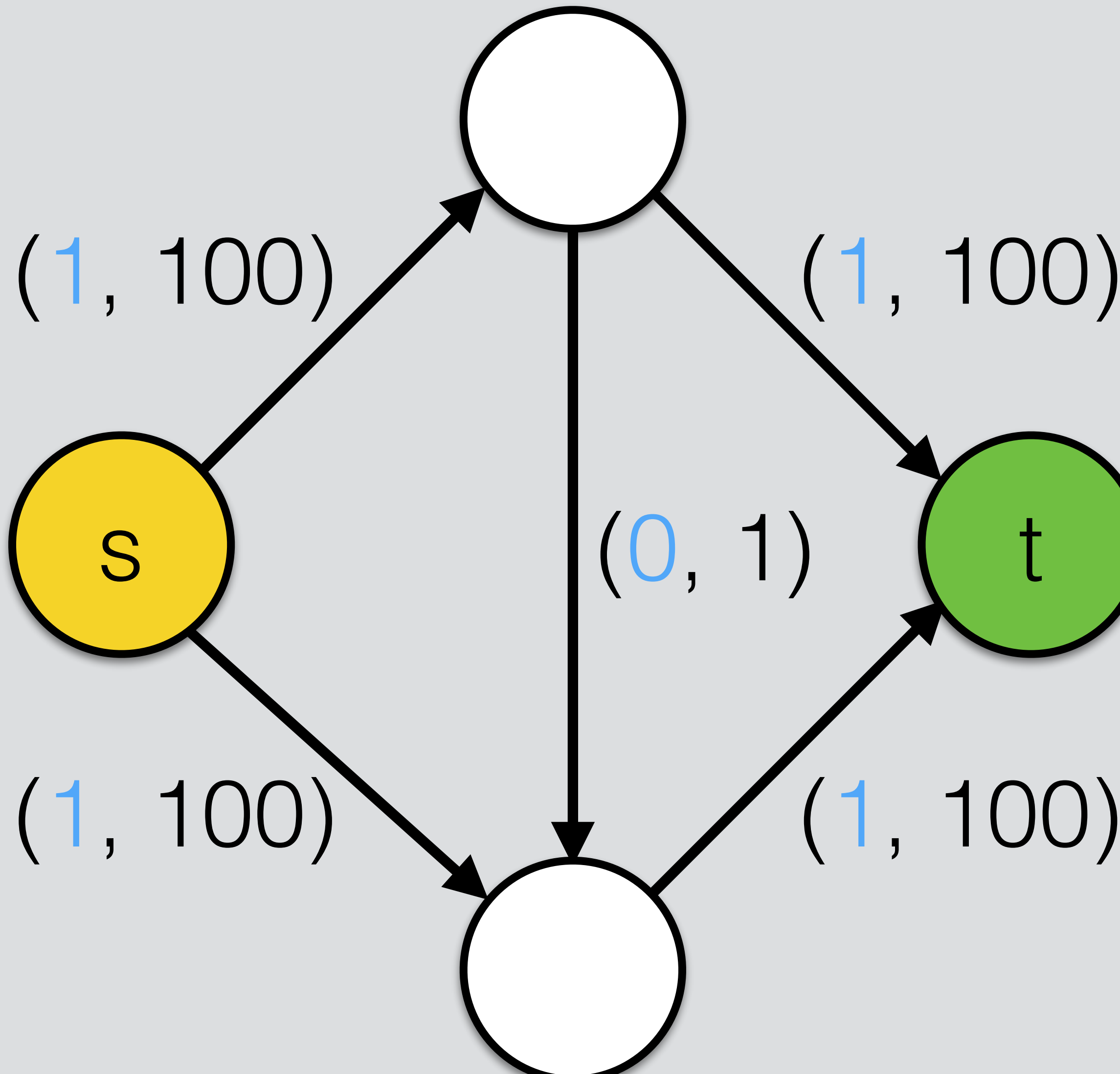
Graph (with flow)



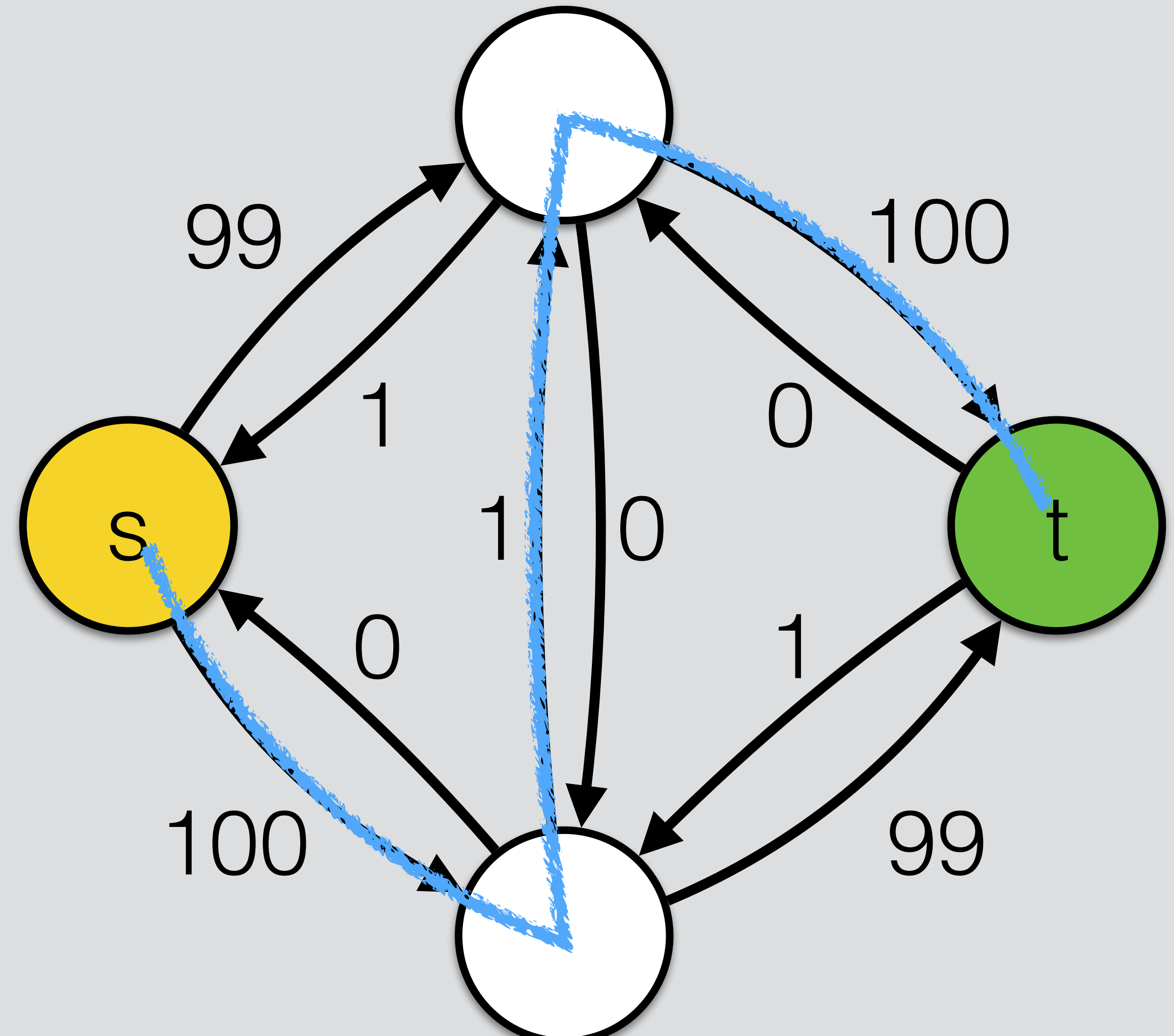
Residual graph



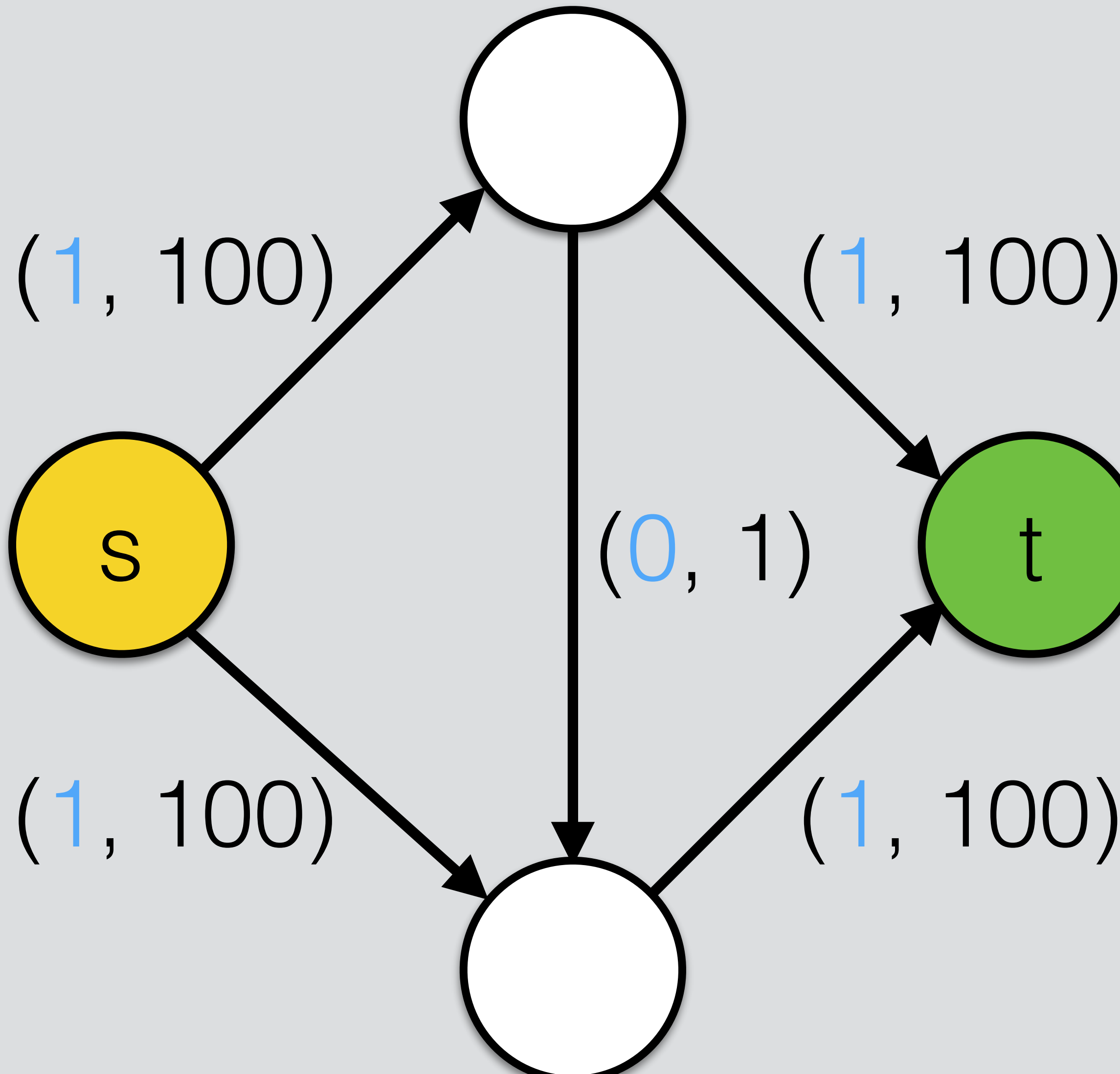
Graph (with flow)



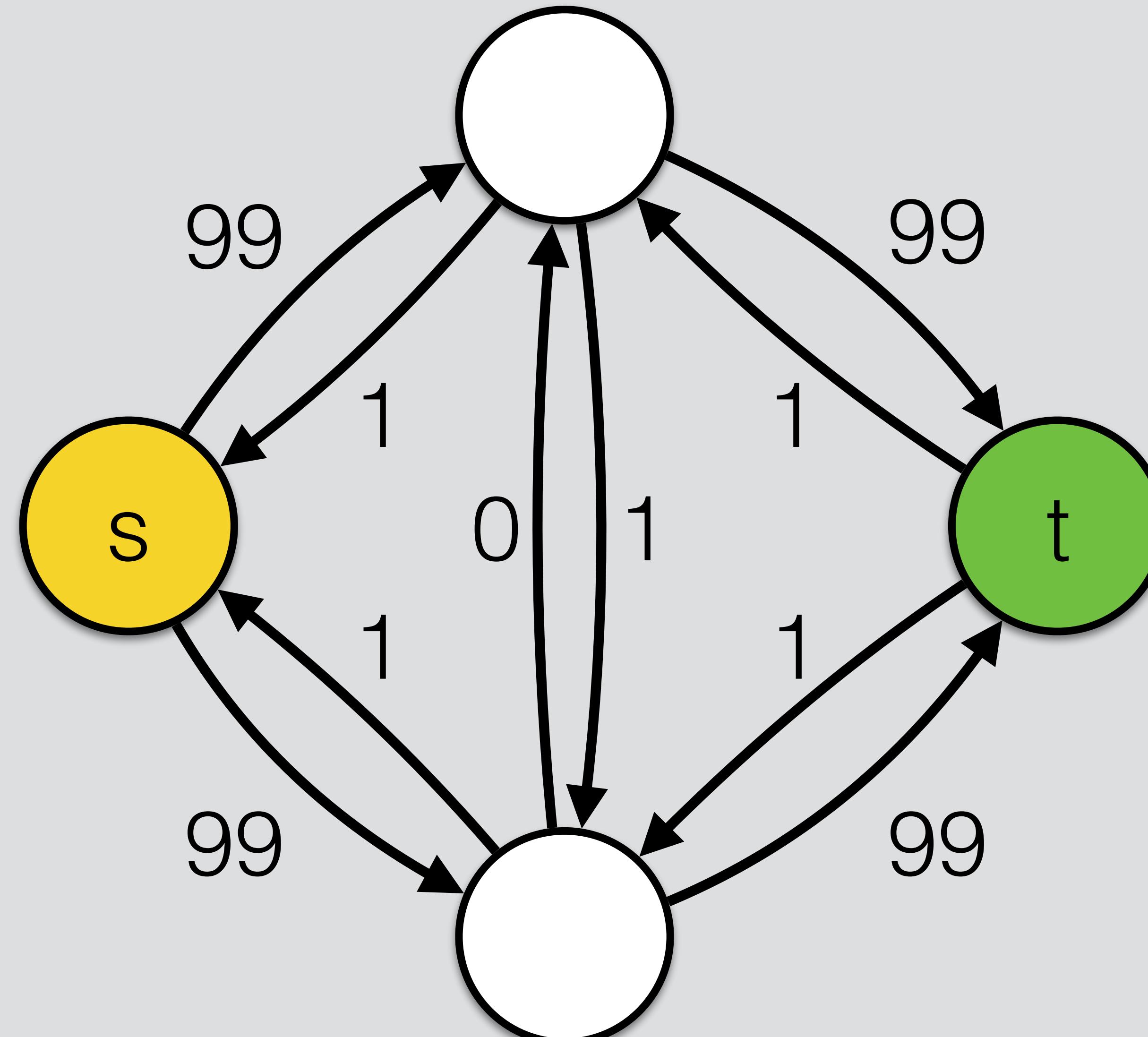
Residual graph



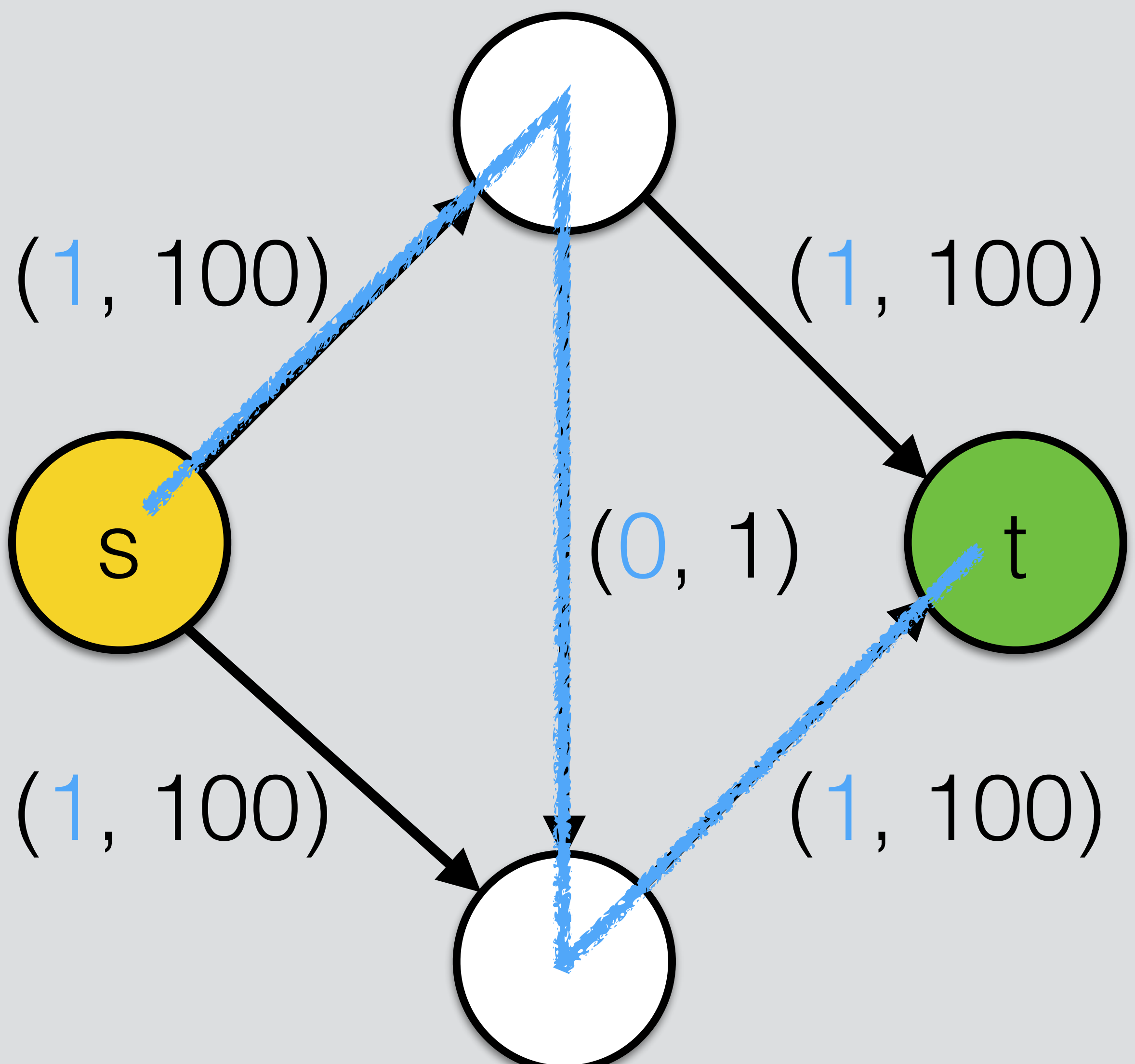
Graph (with flow)



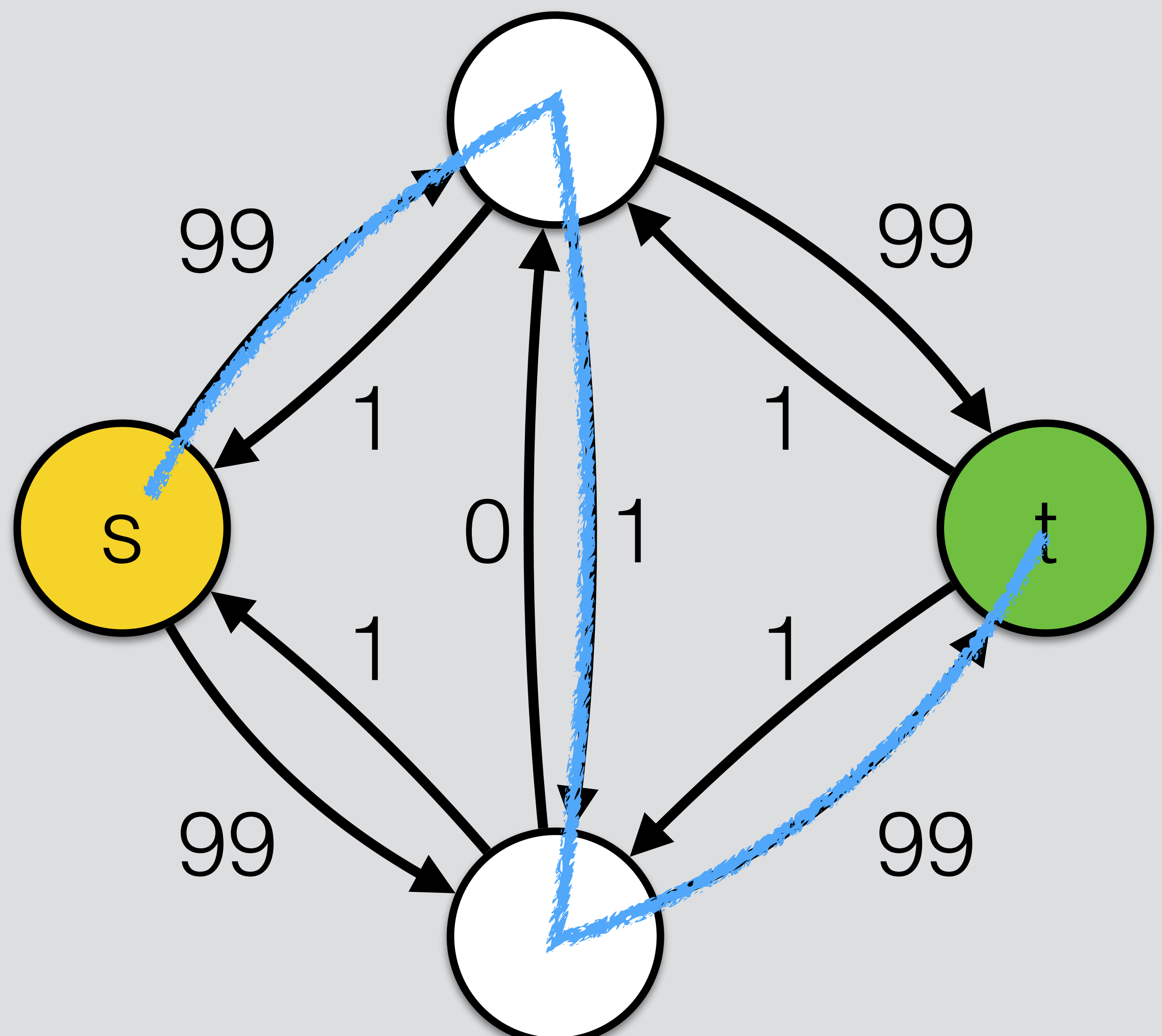
Residual graph



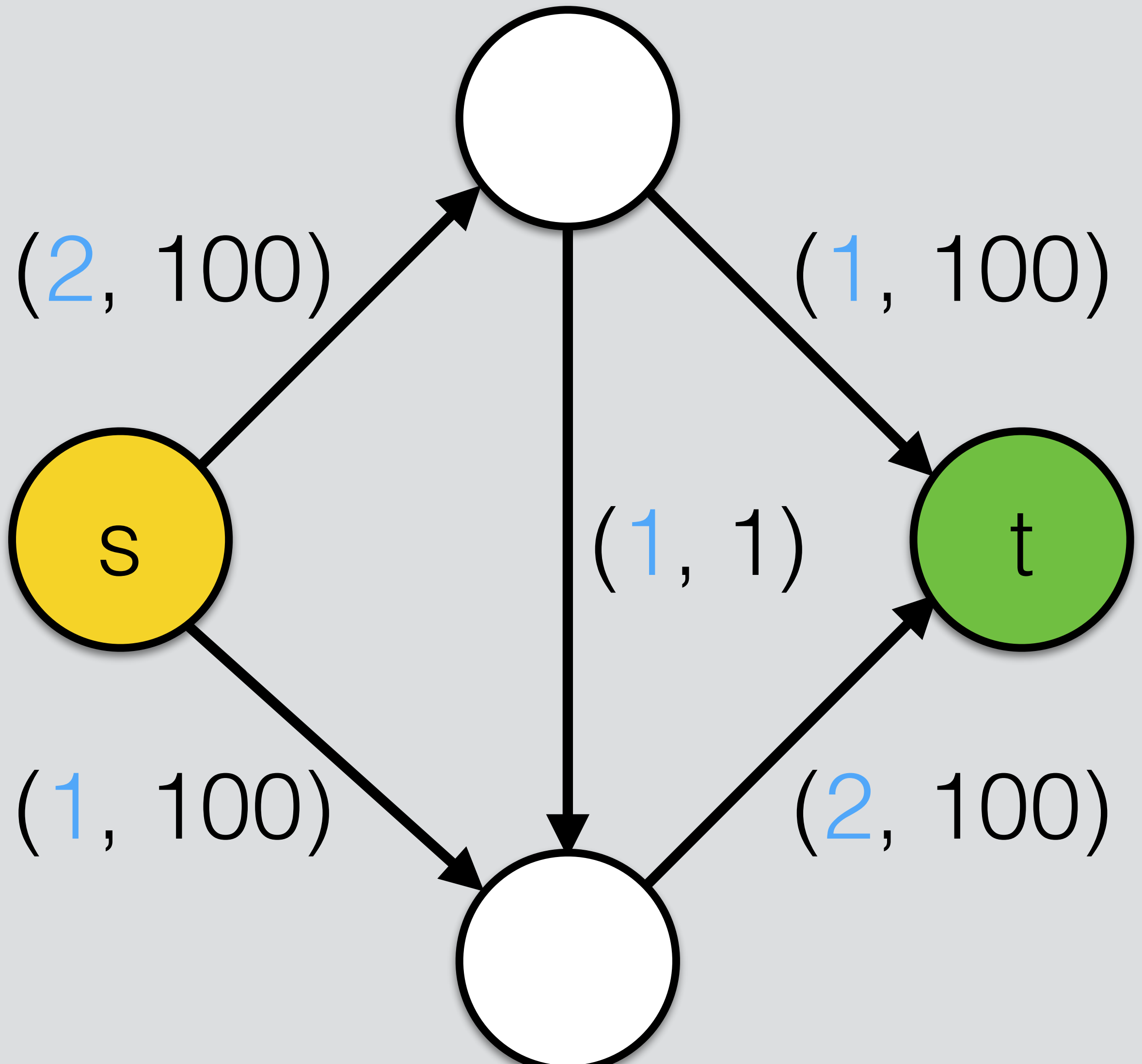
Graph (with flow)



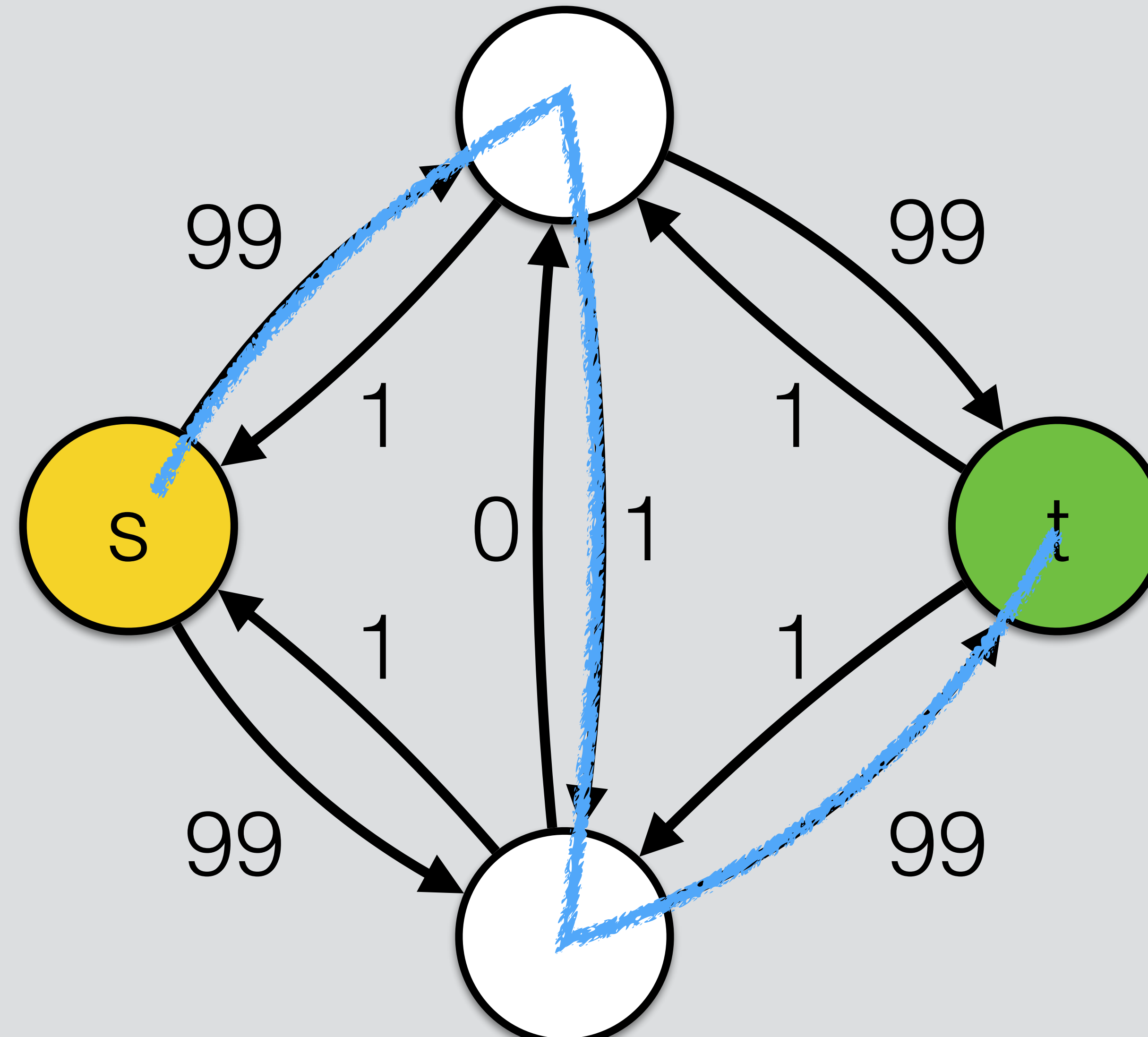
Residual graph



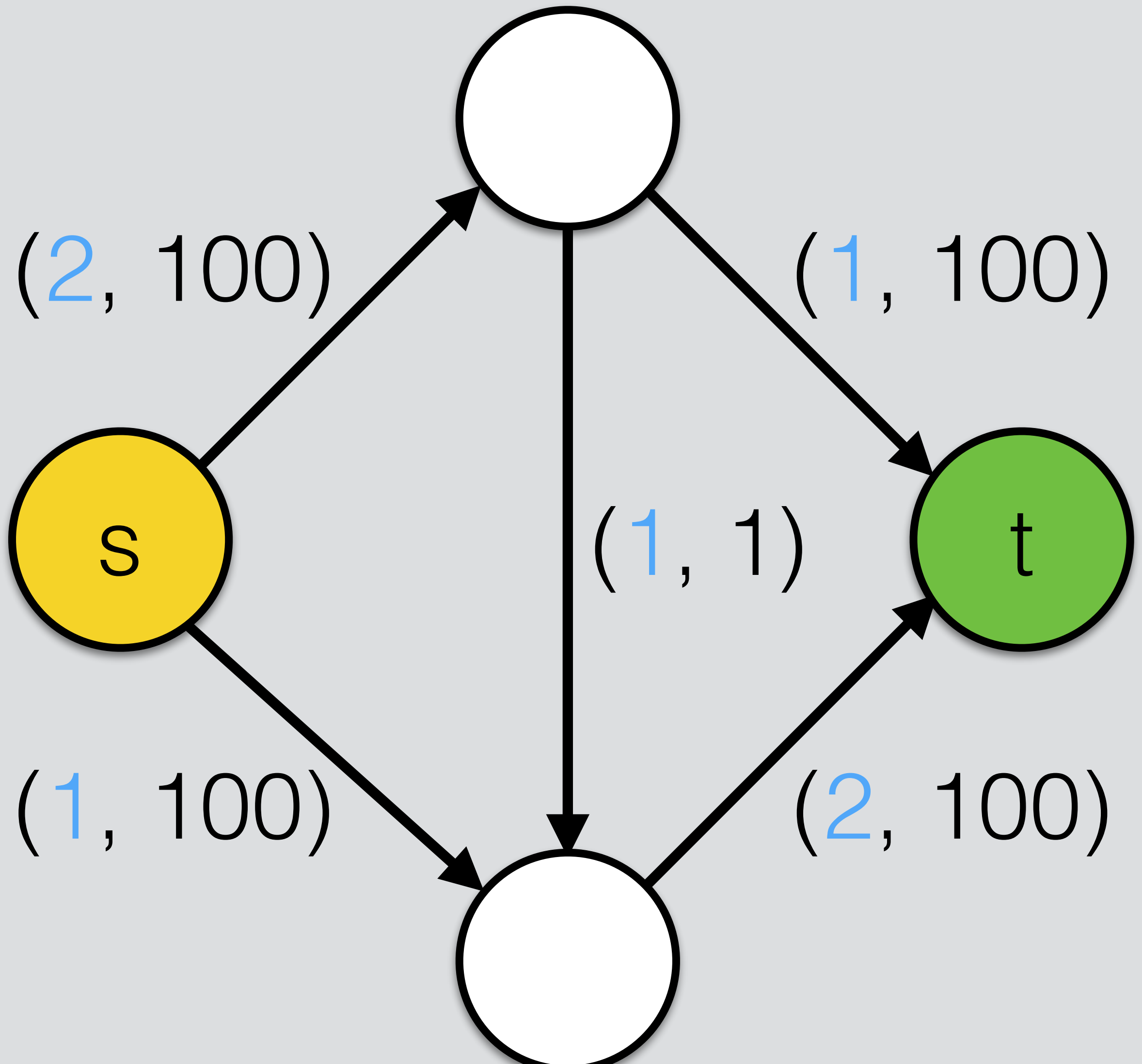
Graph (with flow)



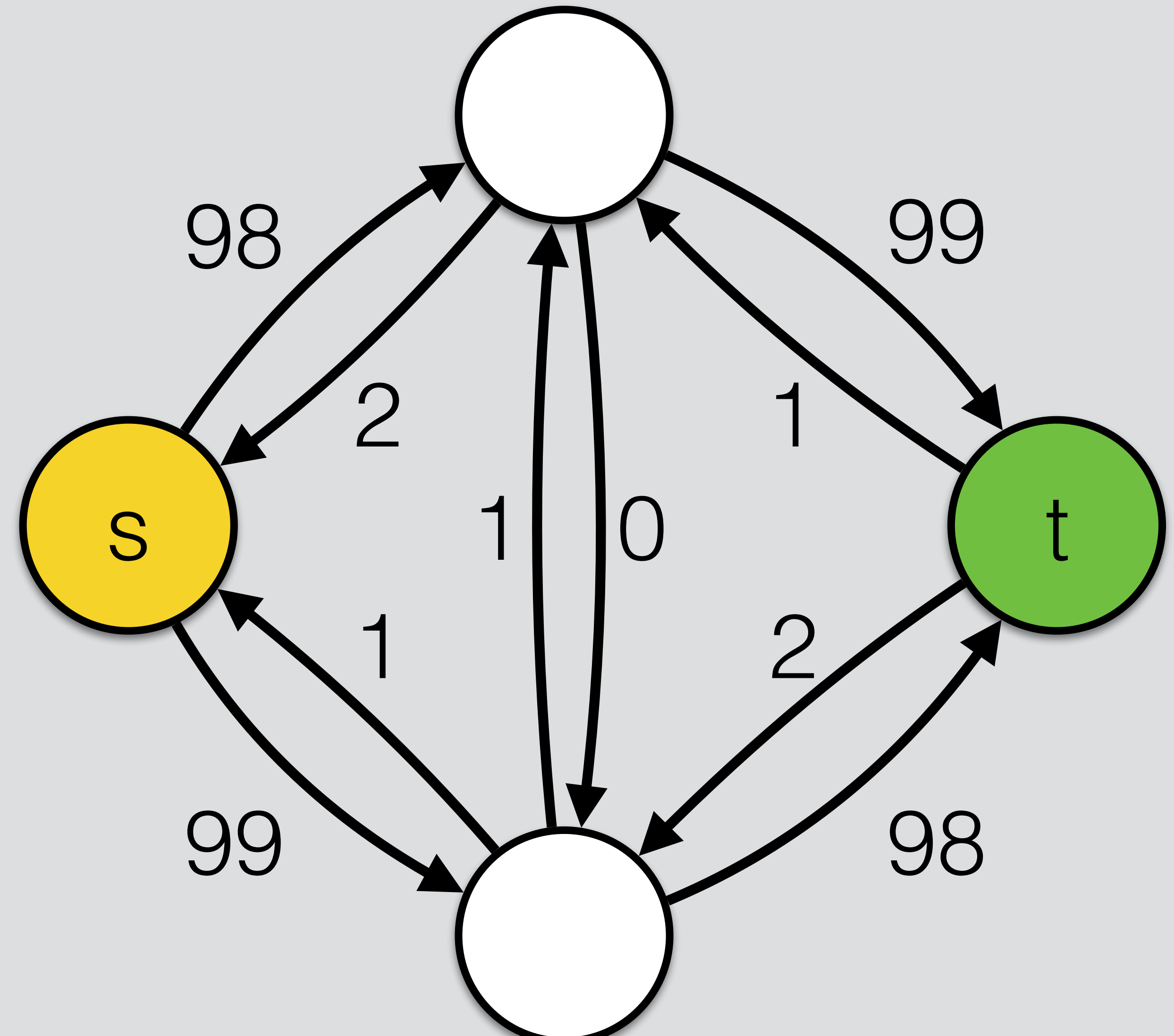
Residual graph



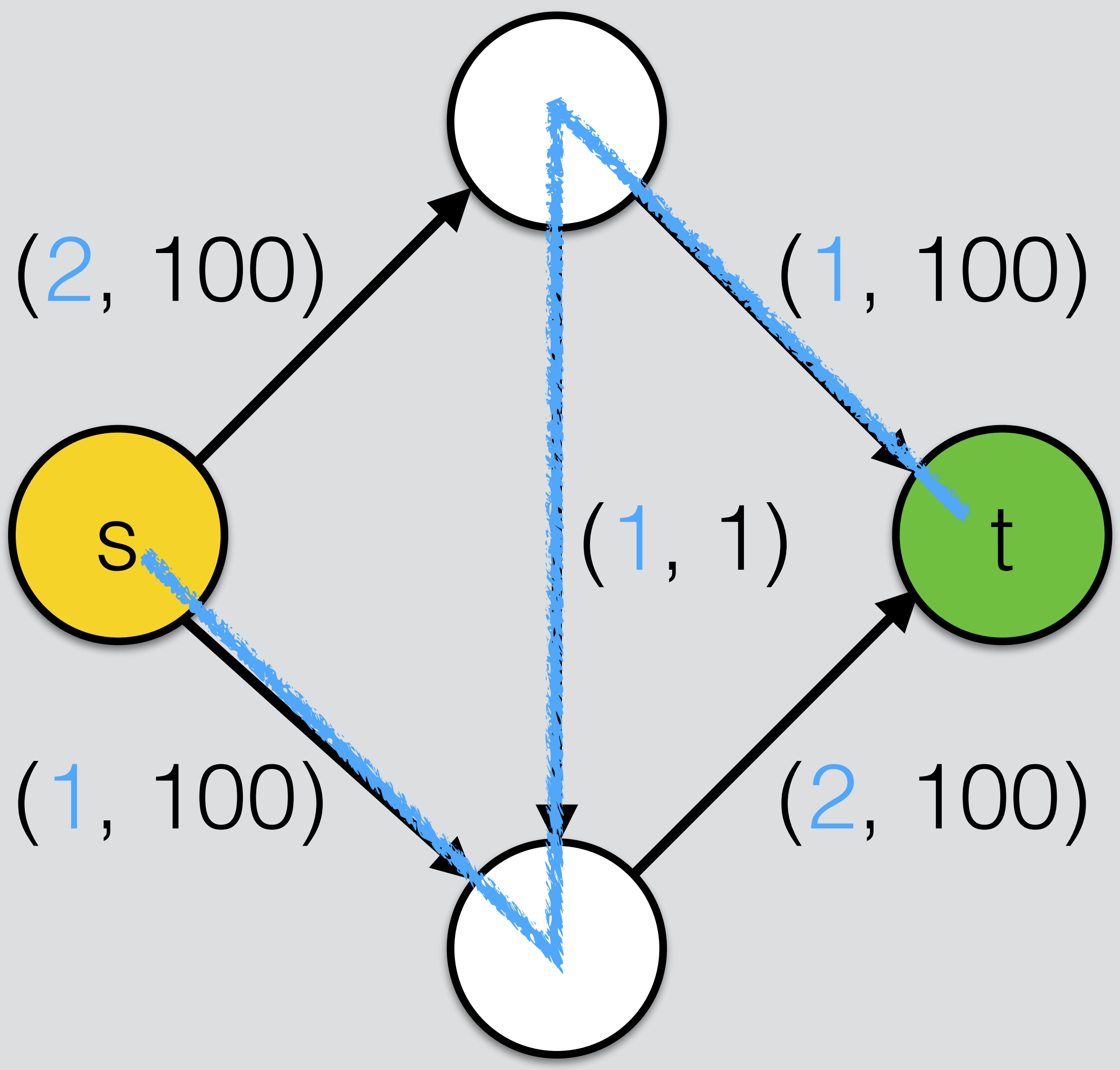
Graph (with flow)



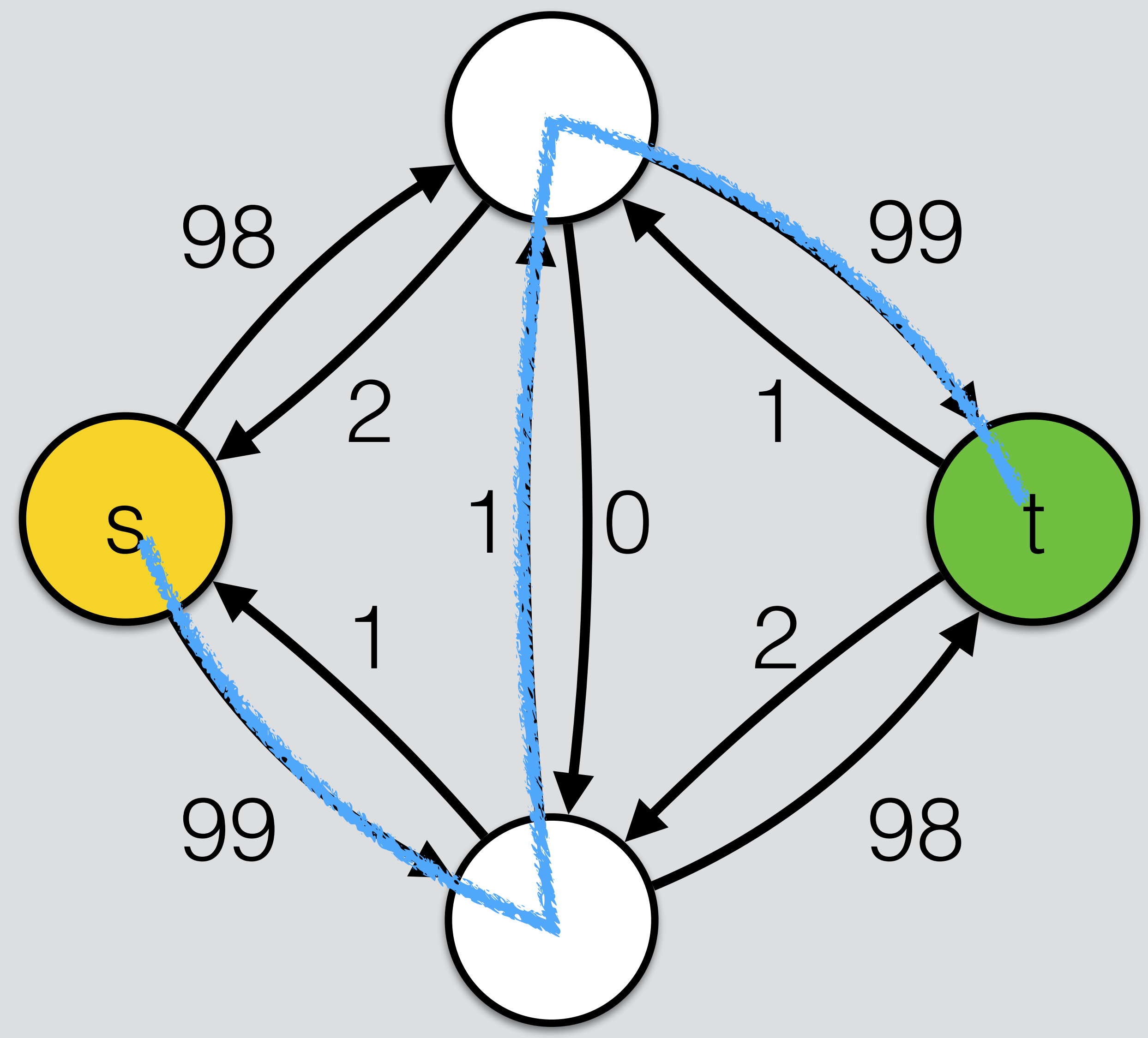
Residual graph

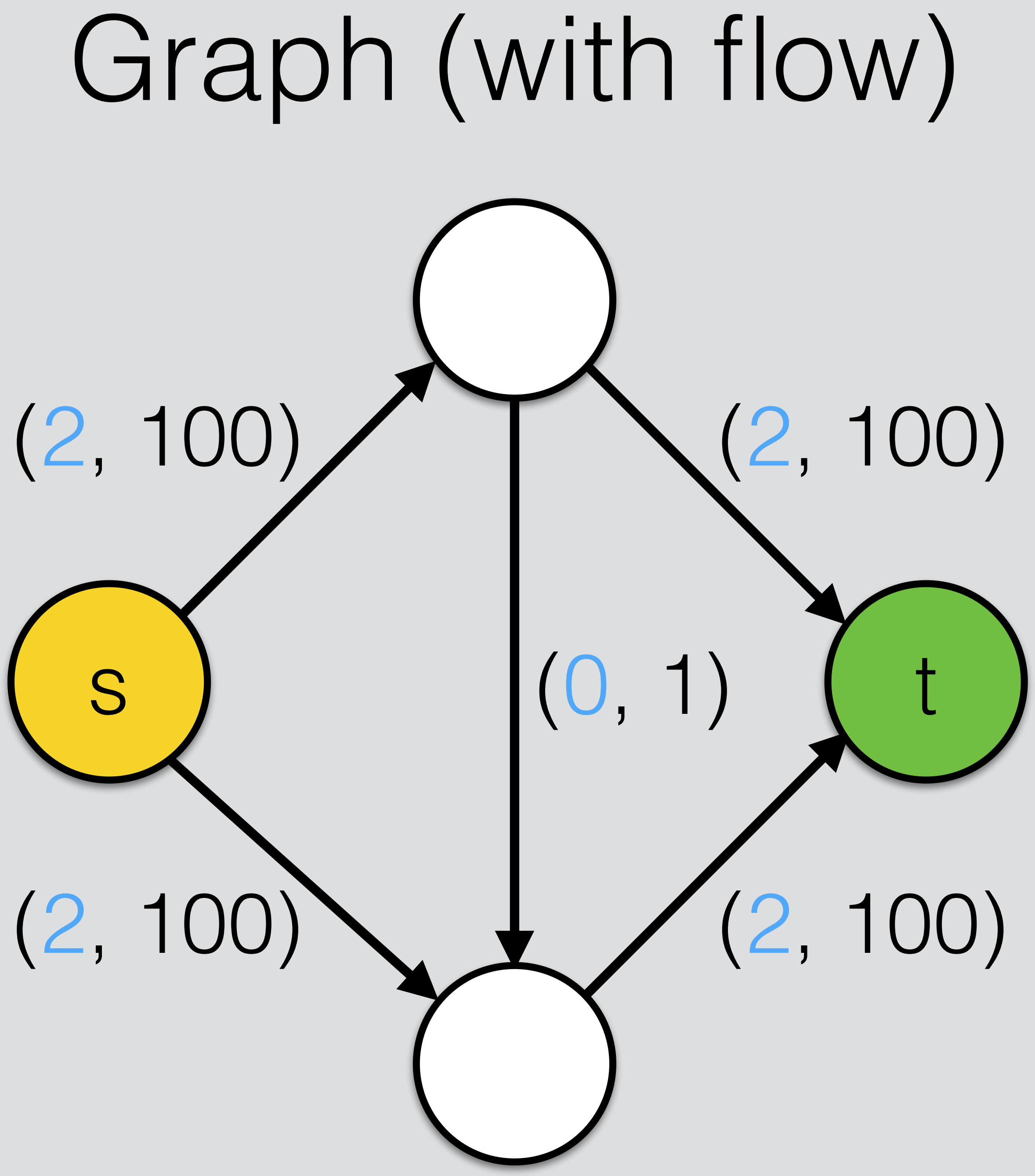


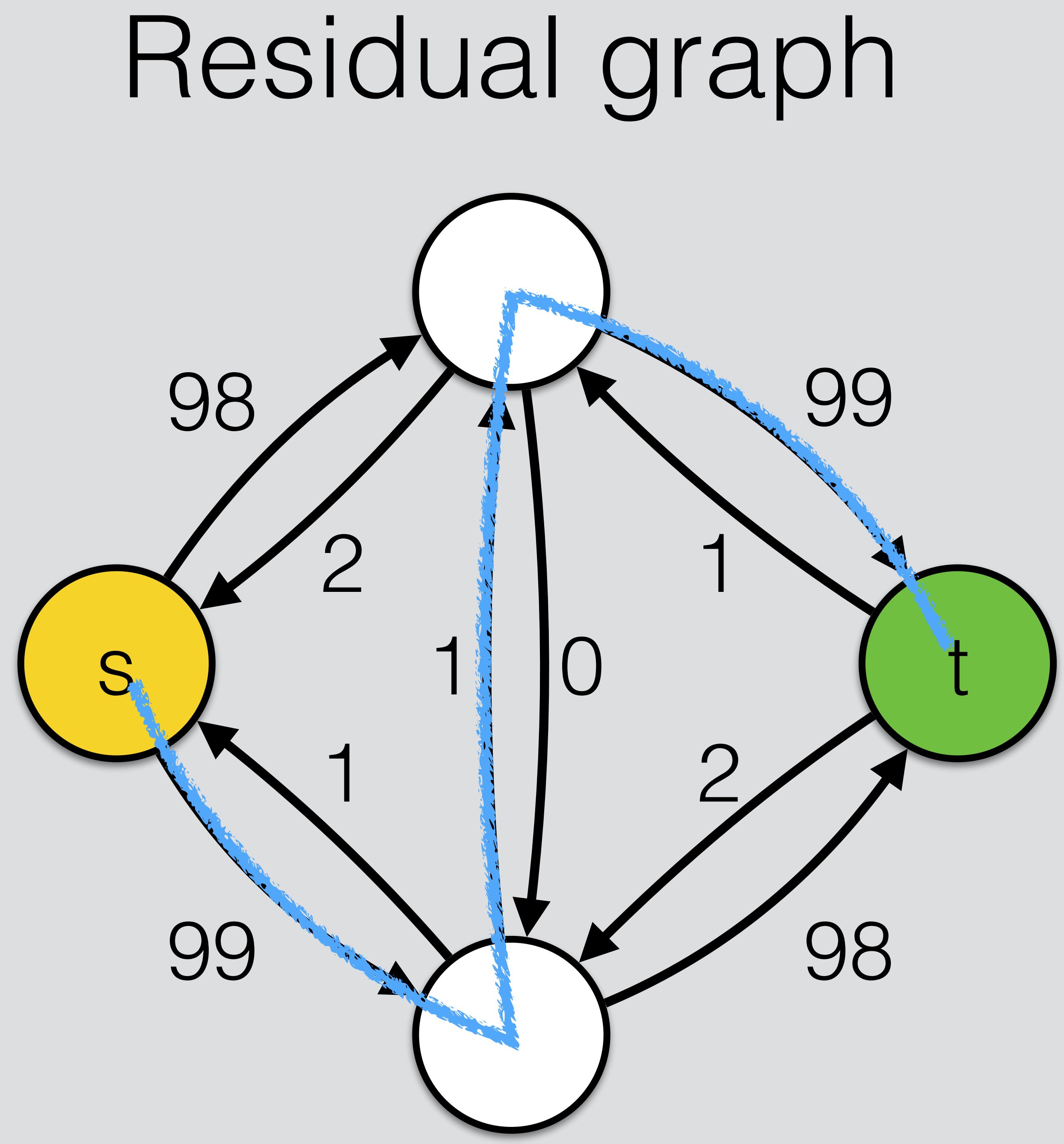
Graph (with flow)



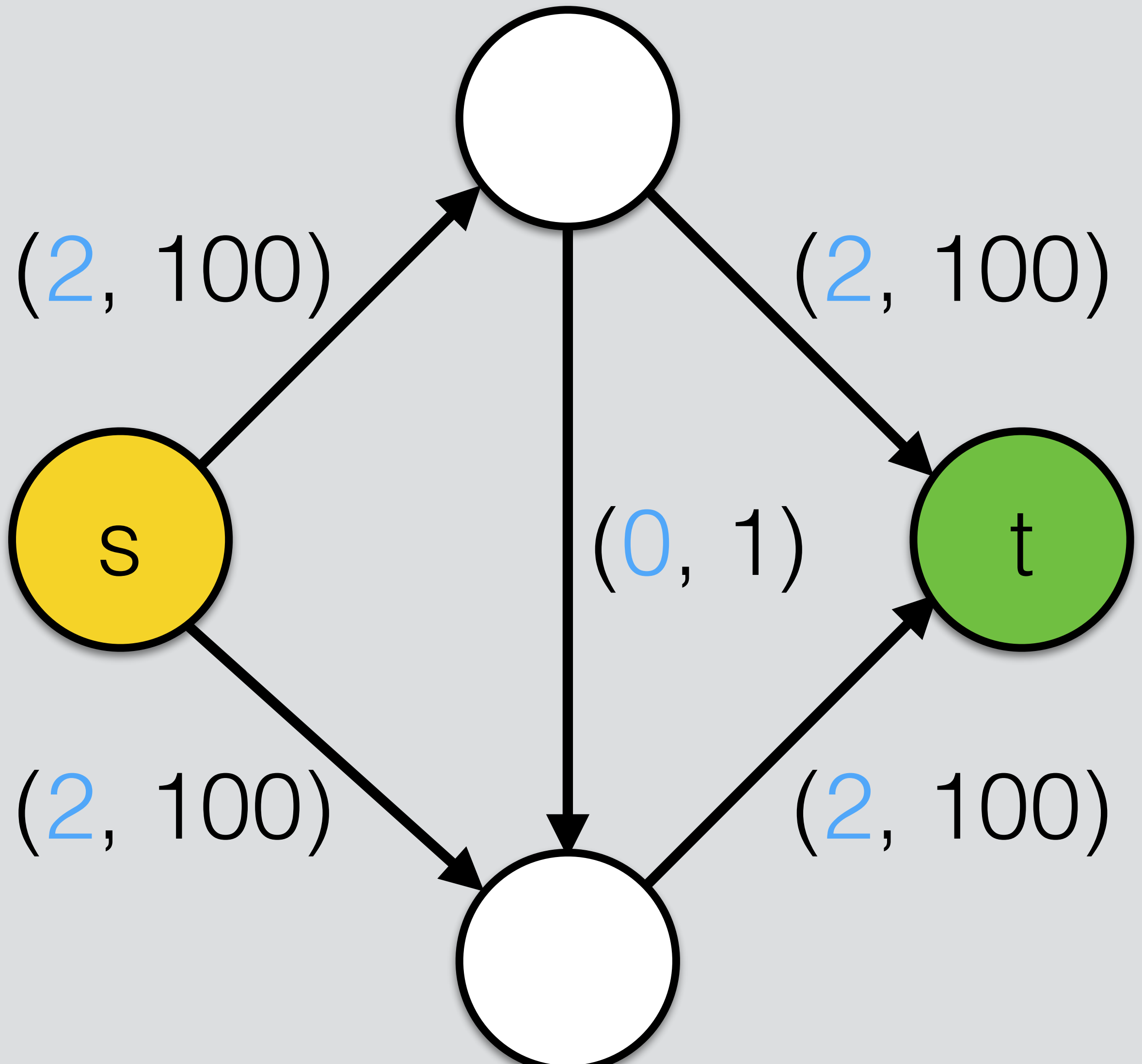
Residual graph



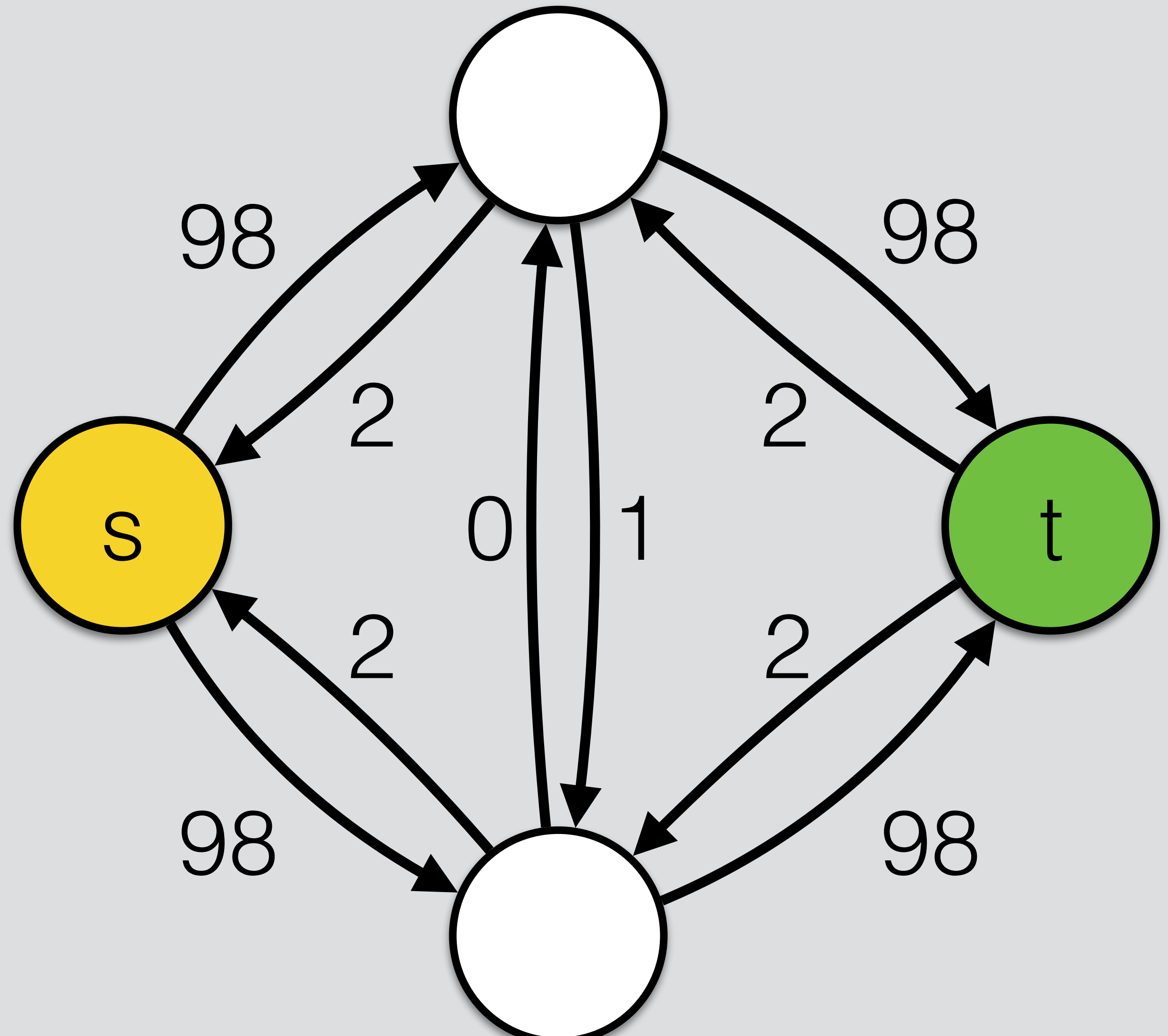

Graph (with flow)
(2, 100)
(2, 100)
(0, 1)
s
t
(2, 100)
(2, 100)


Residual graph
98
99
2
1
1
0
s
t
1
2
99
98

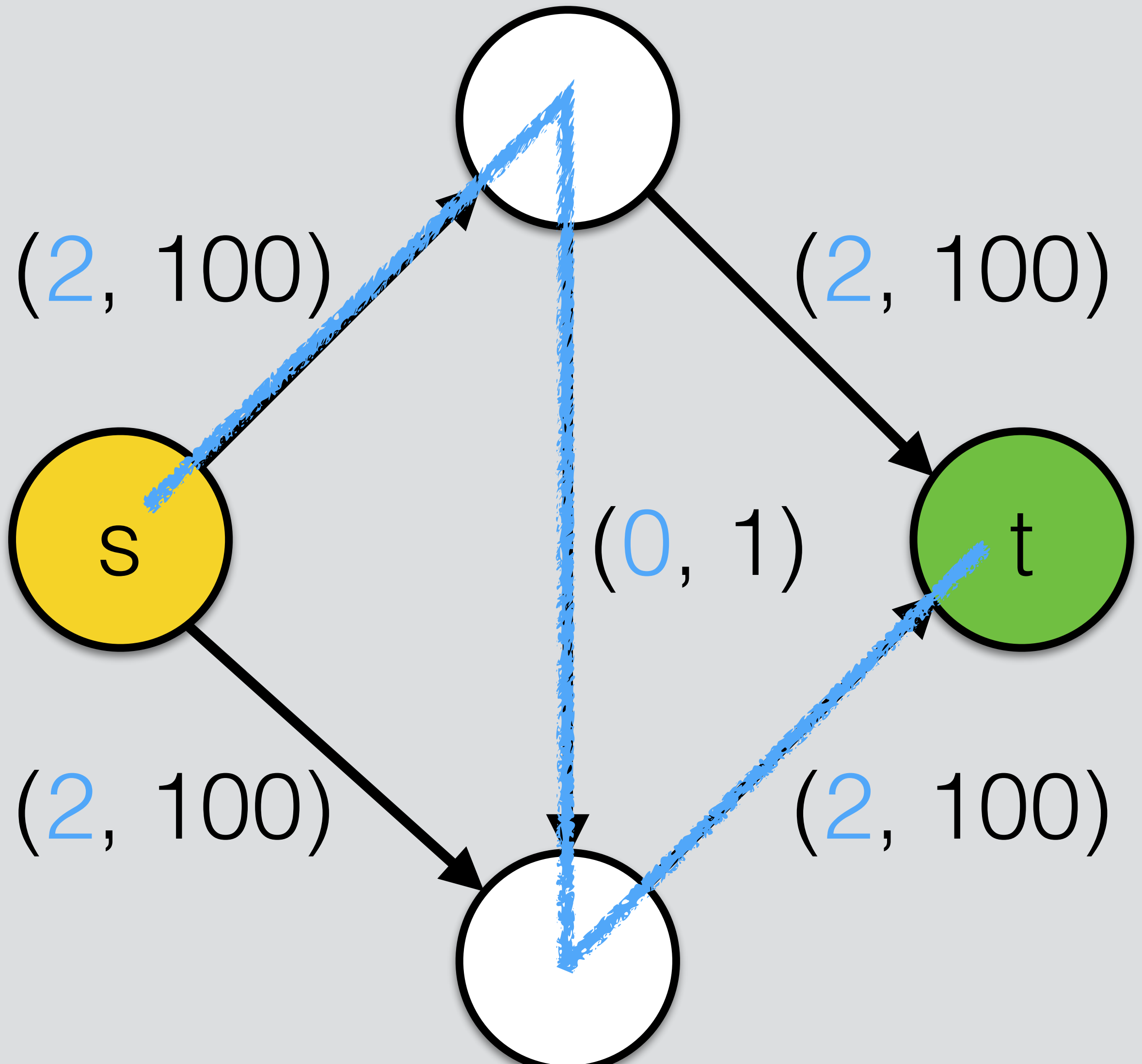
Graph (with flow)



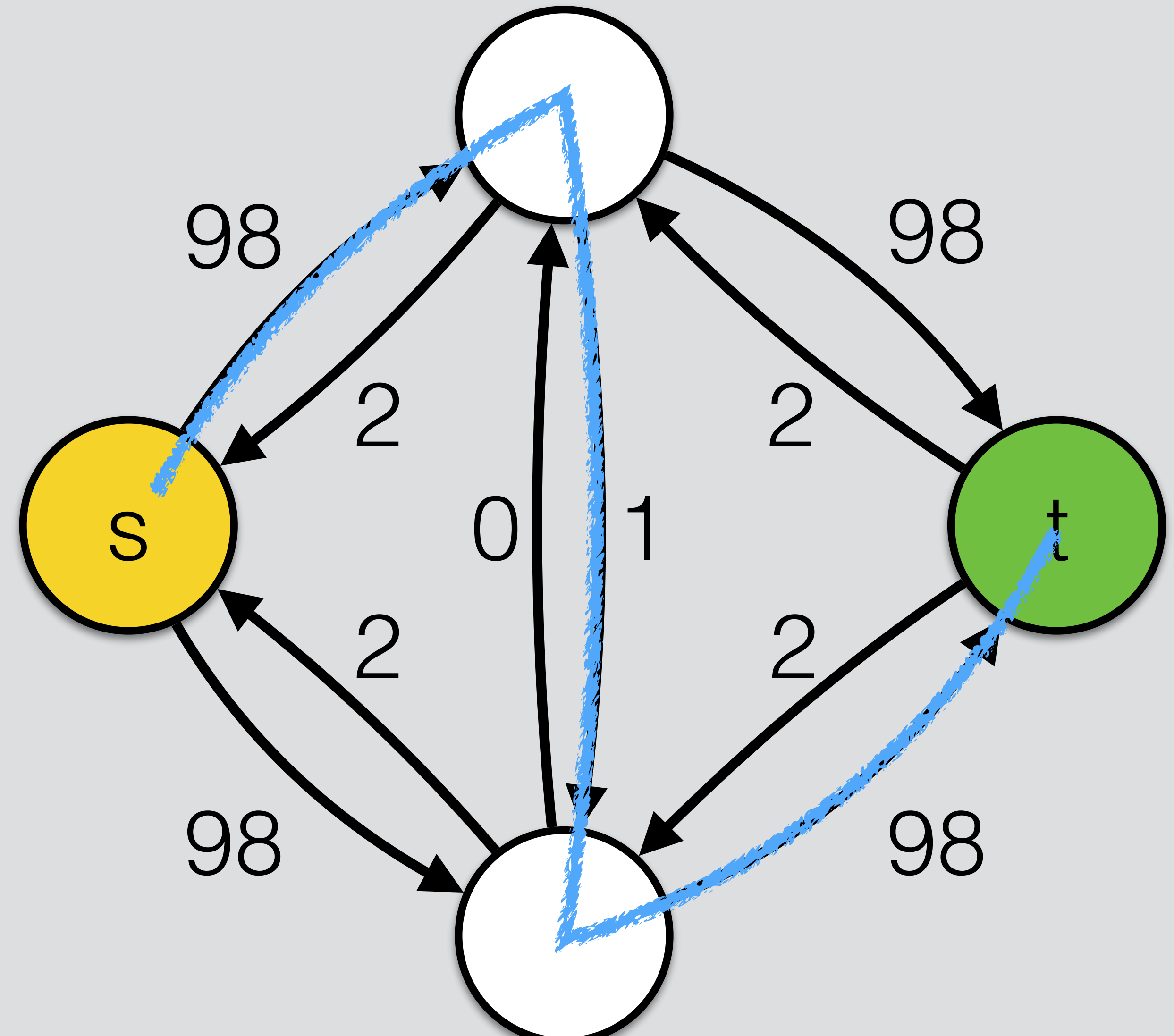
Residual graph



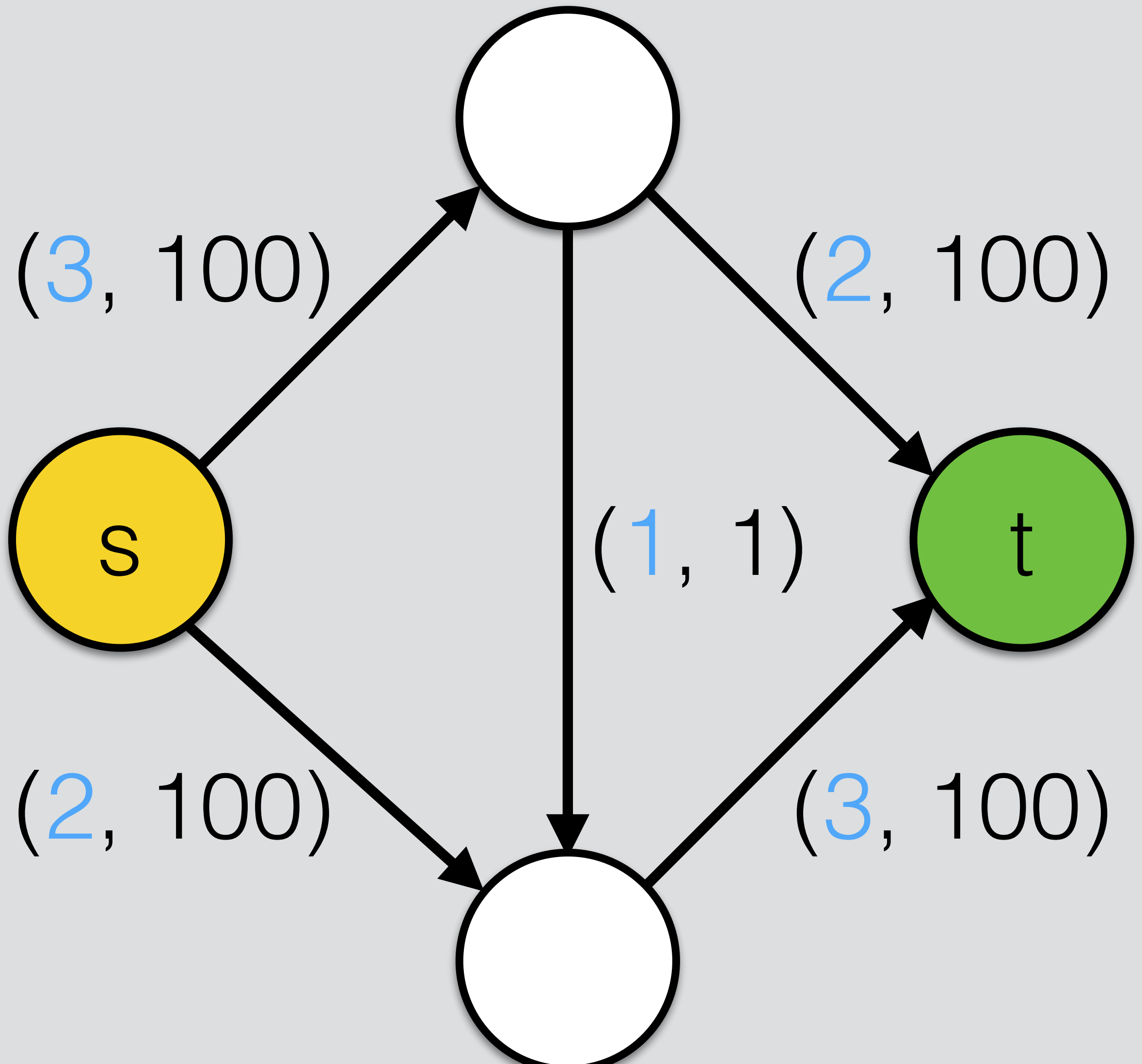
Graph (with flow)



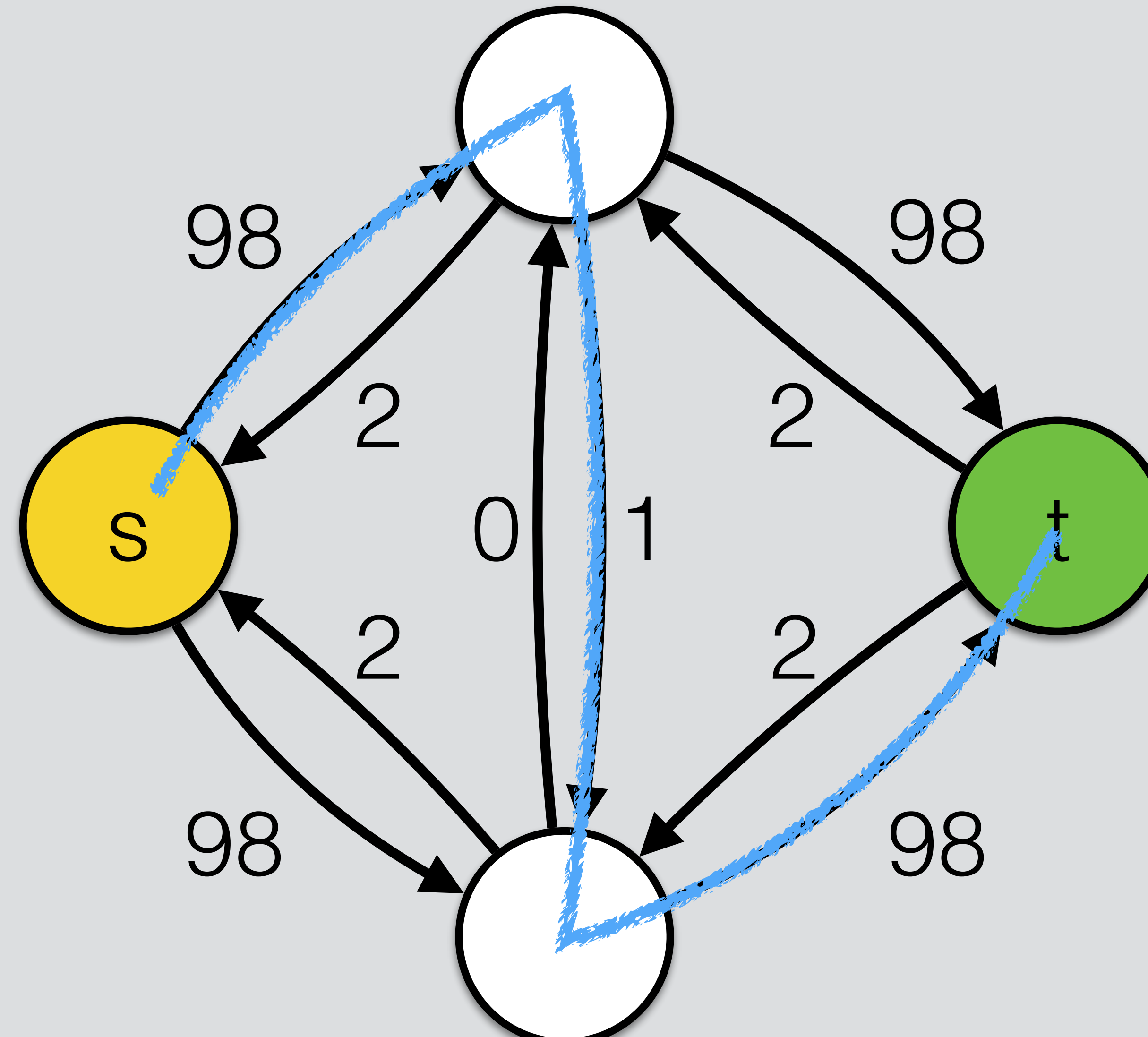
Residual graph



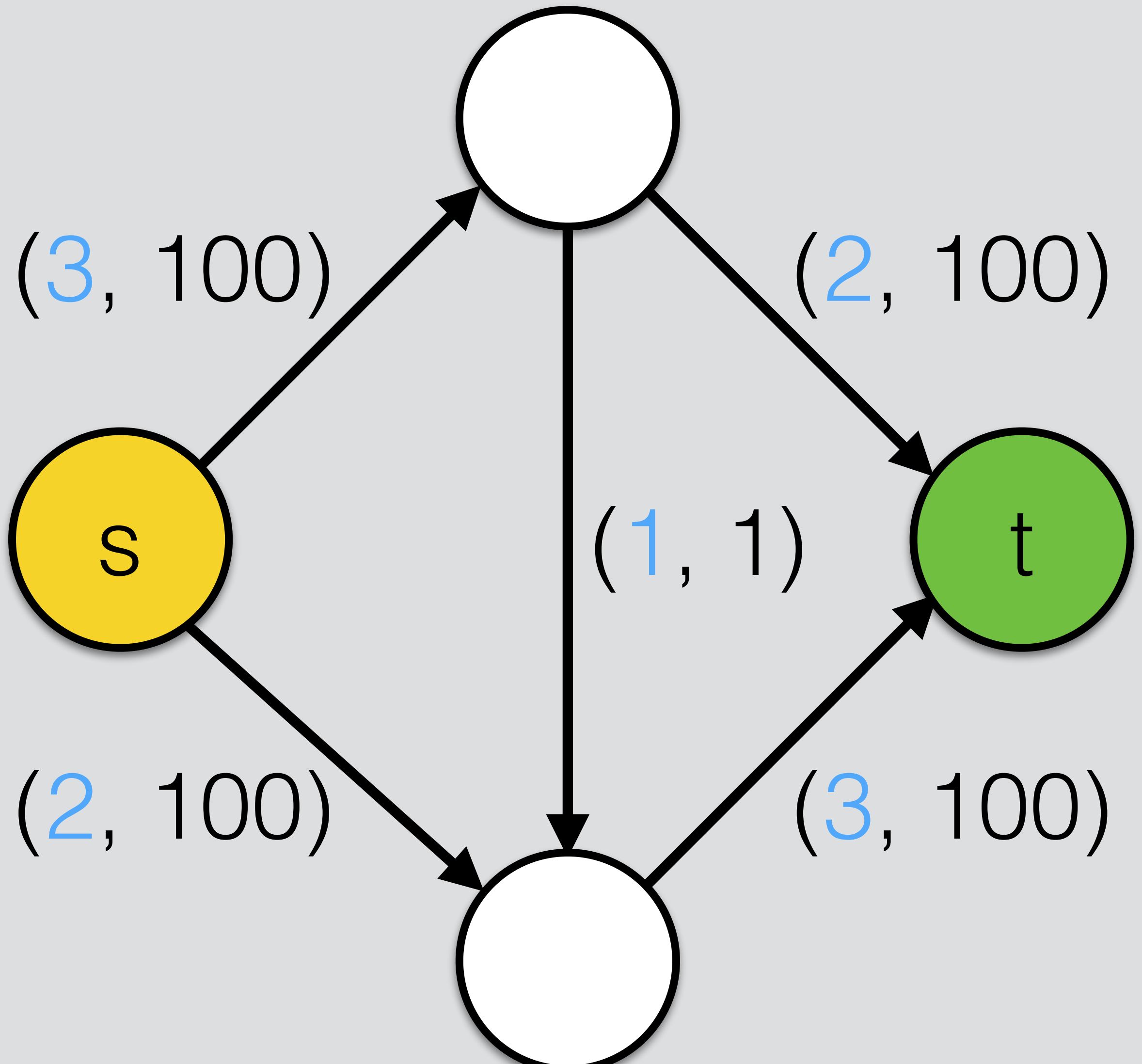
Graph (with flow)



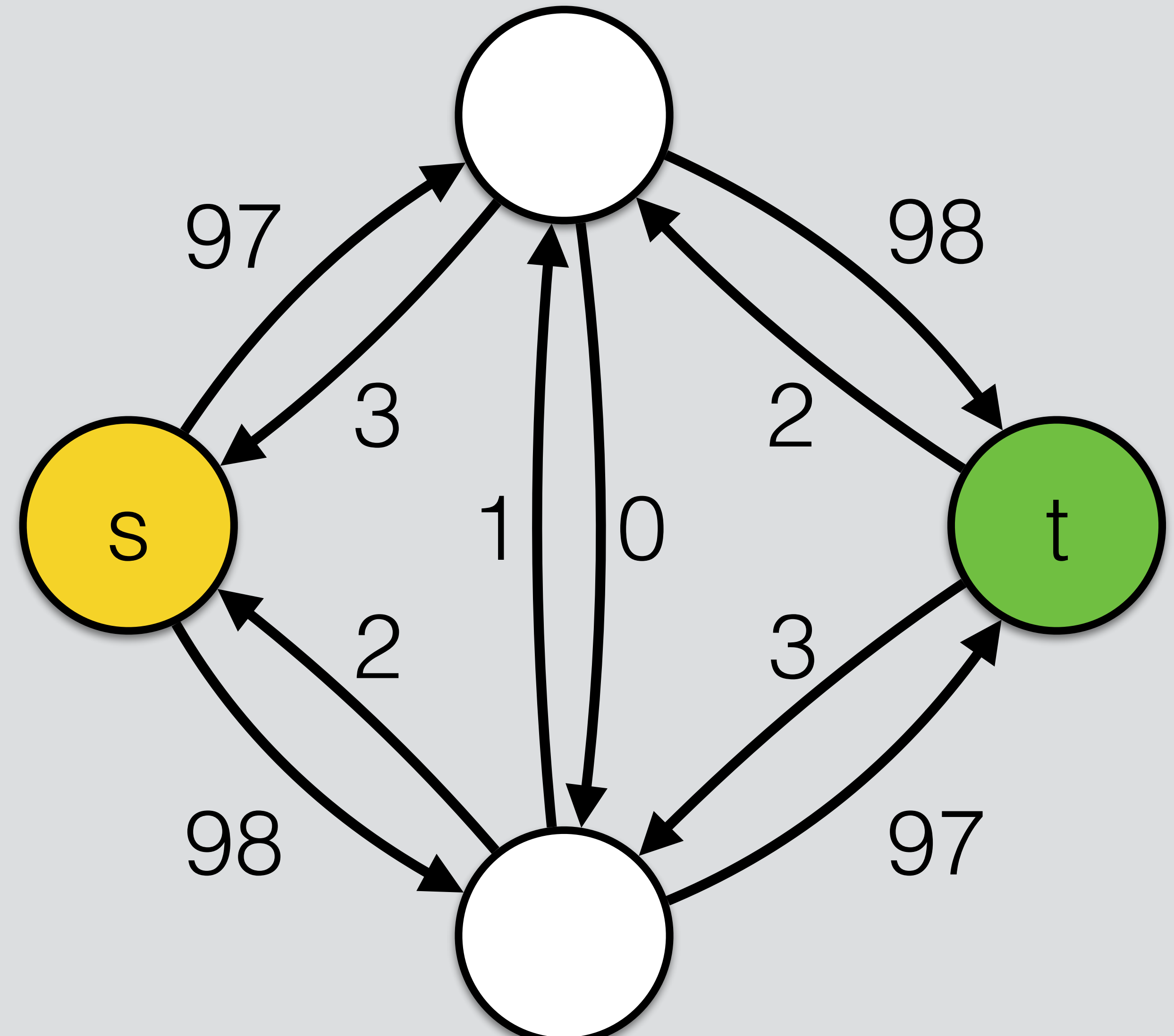
Residual graph



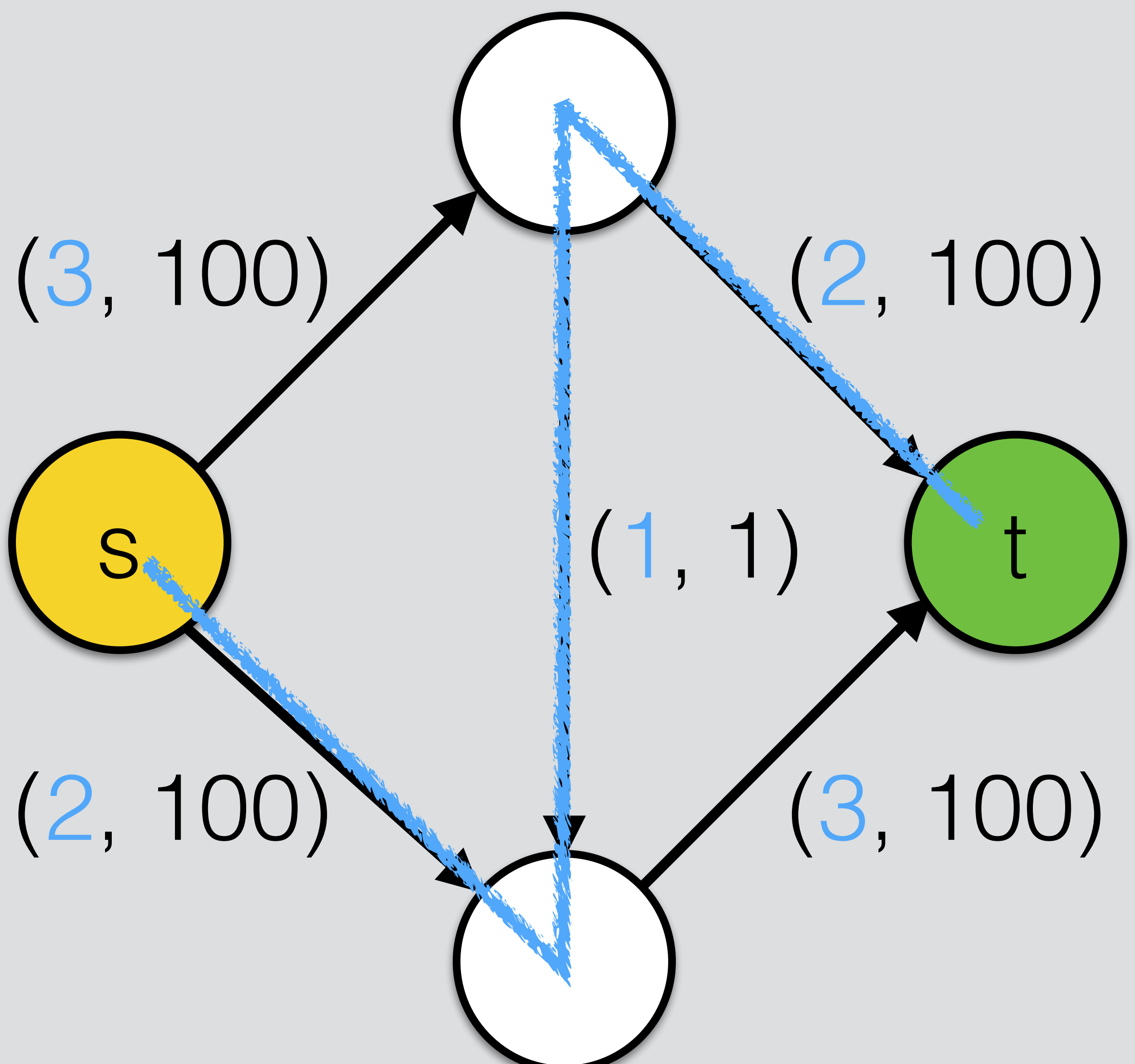
Graph (with flow)



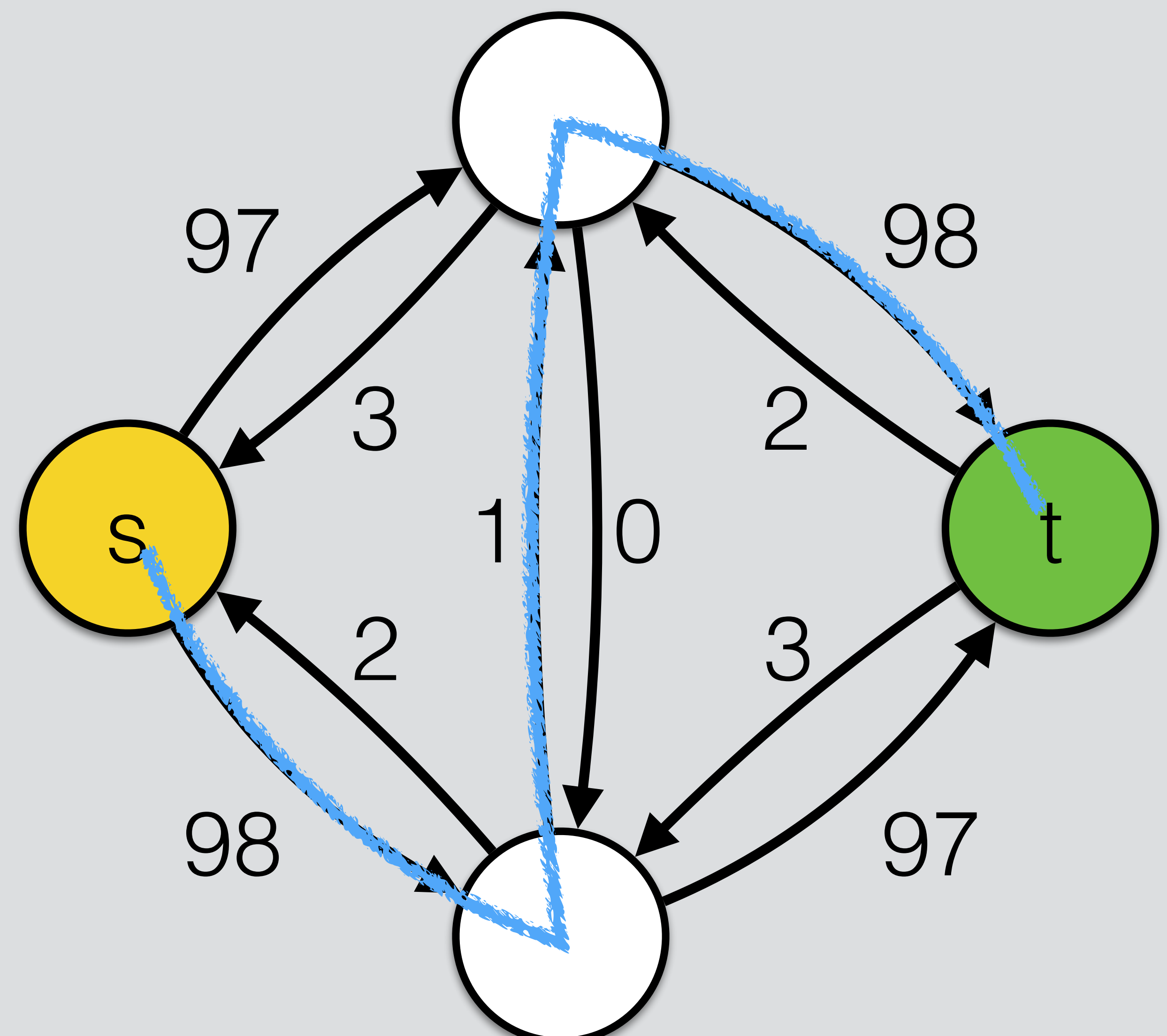
Residual graph



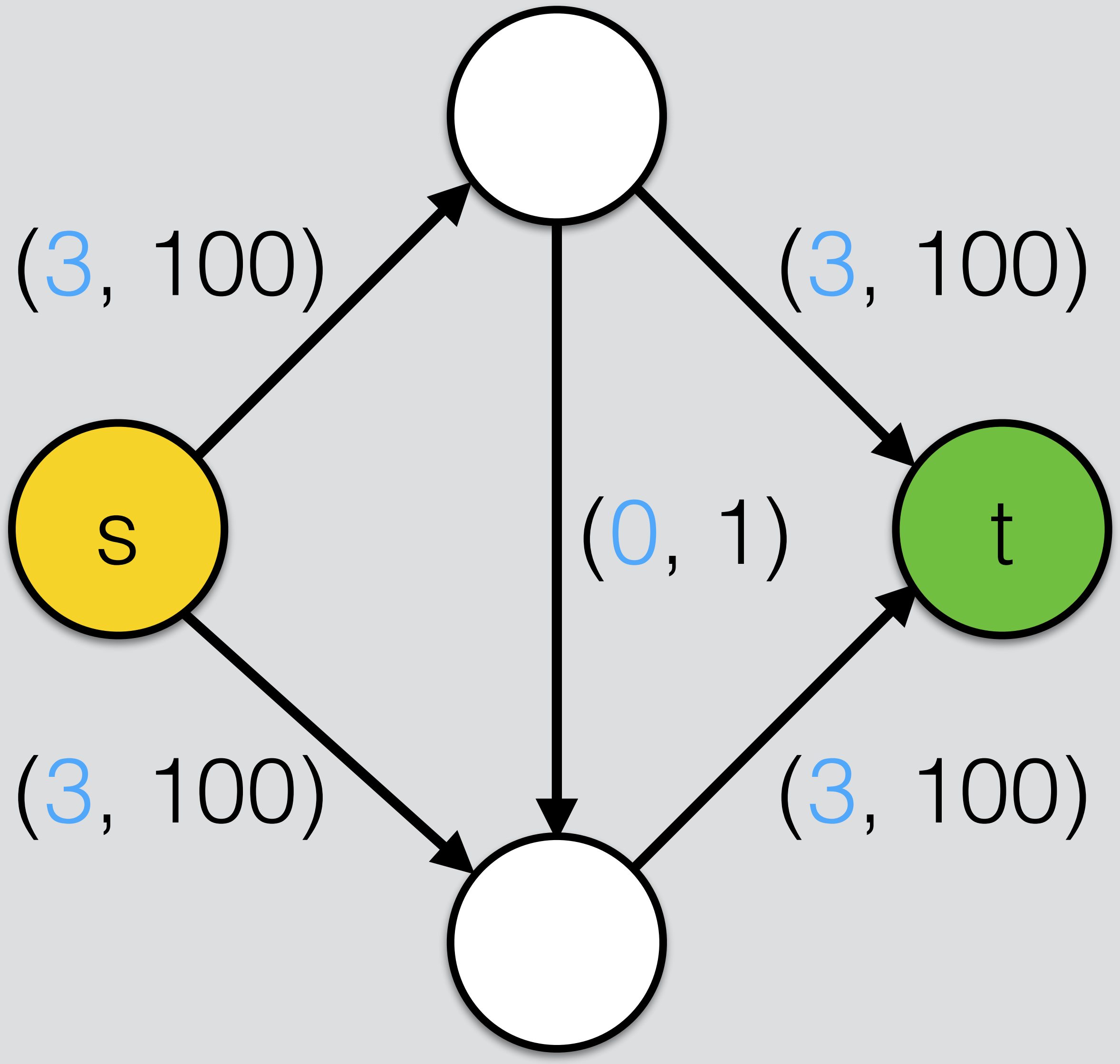
Graph (with flow)



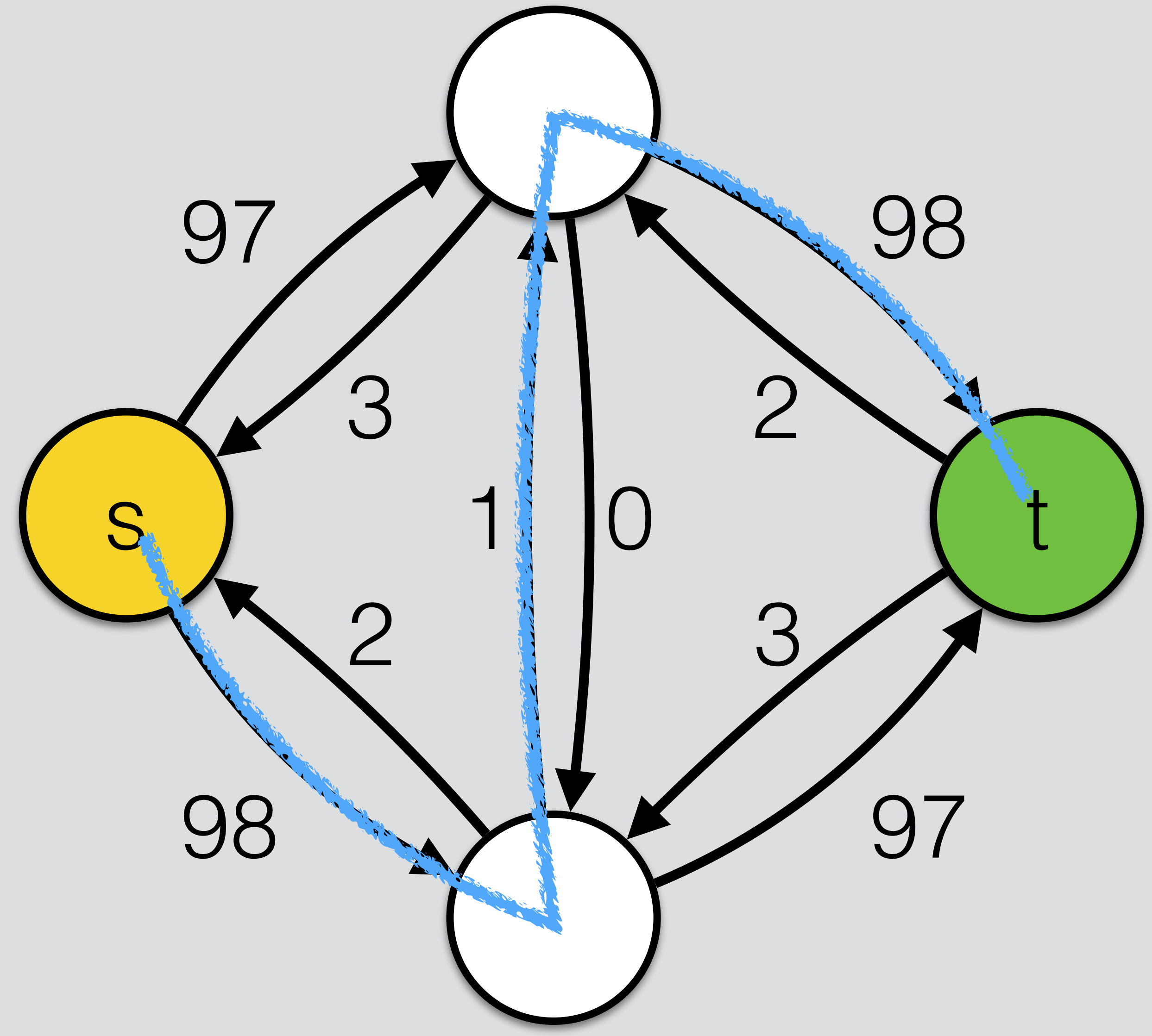
Residual graph



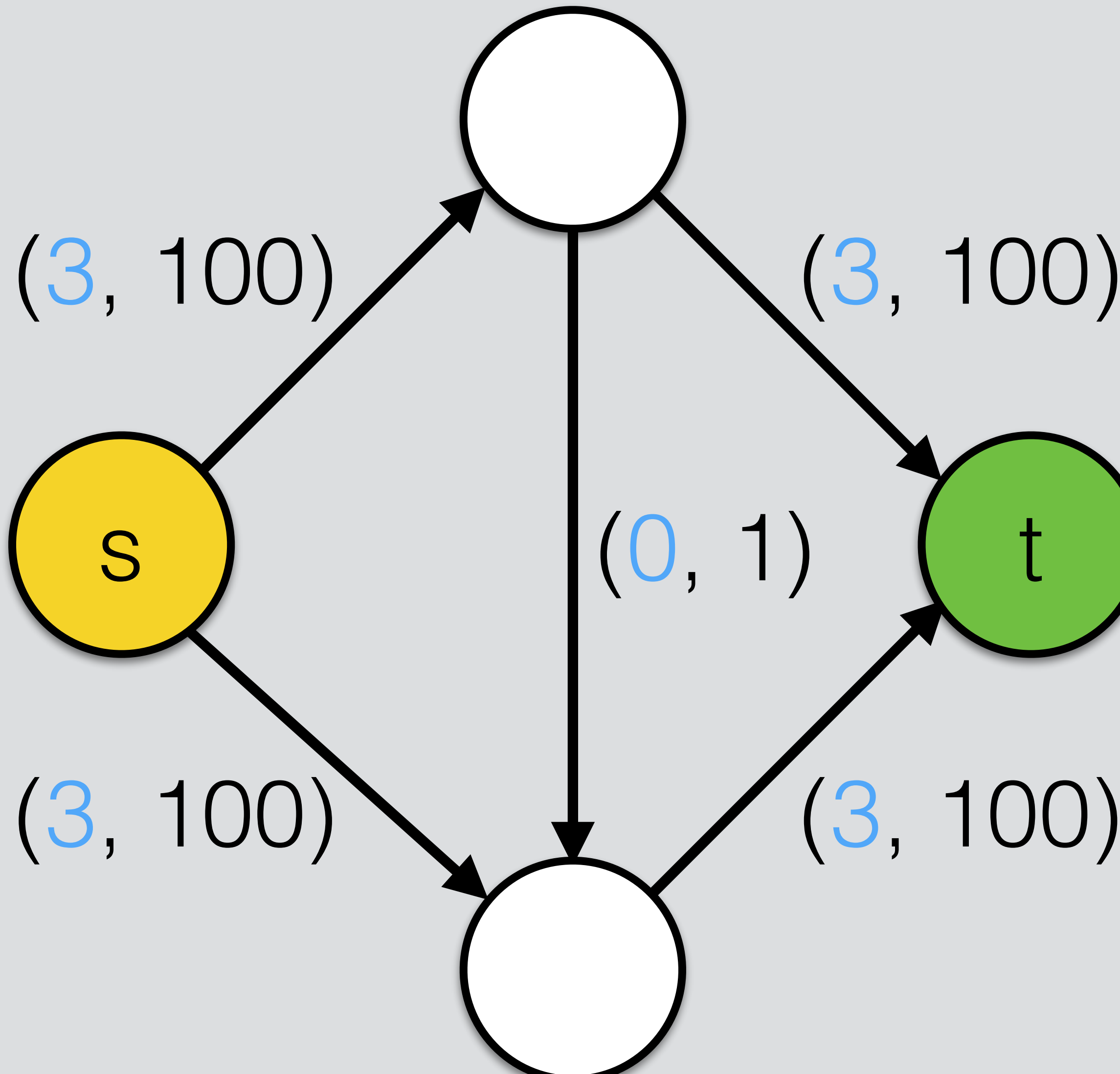
Graph (with flow)



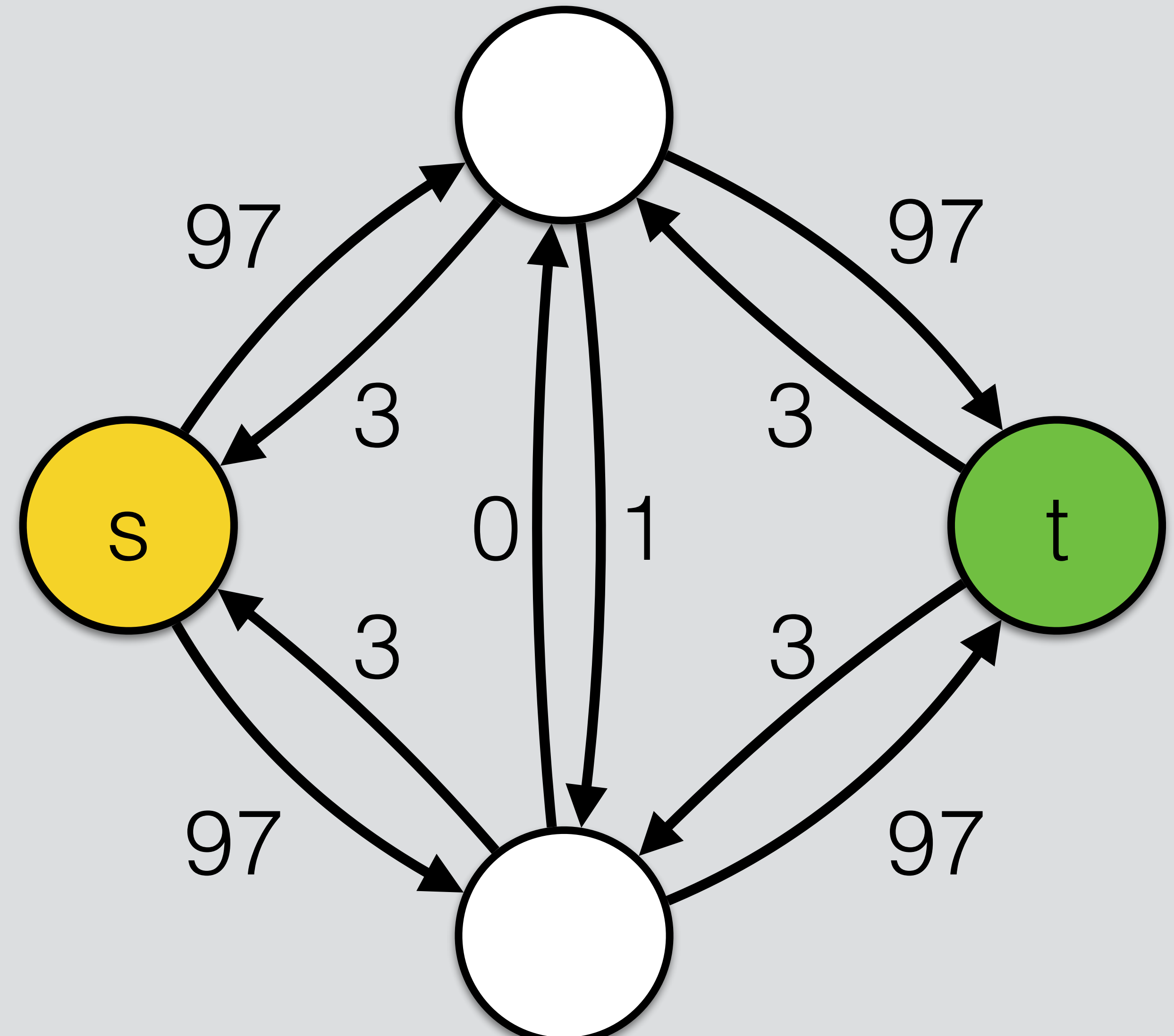
Residual graph



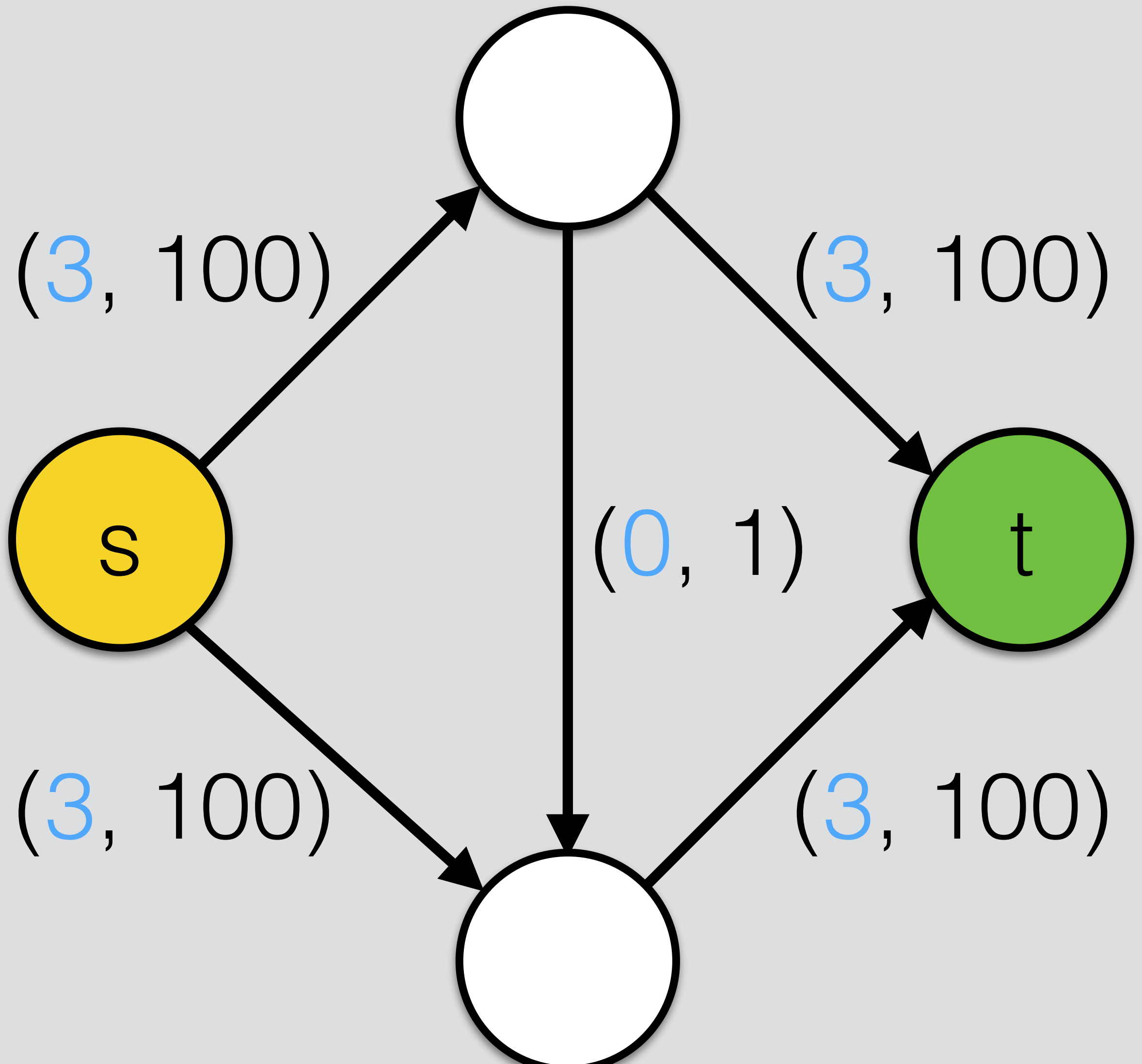
Graph (with flow)



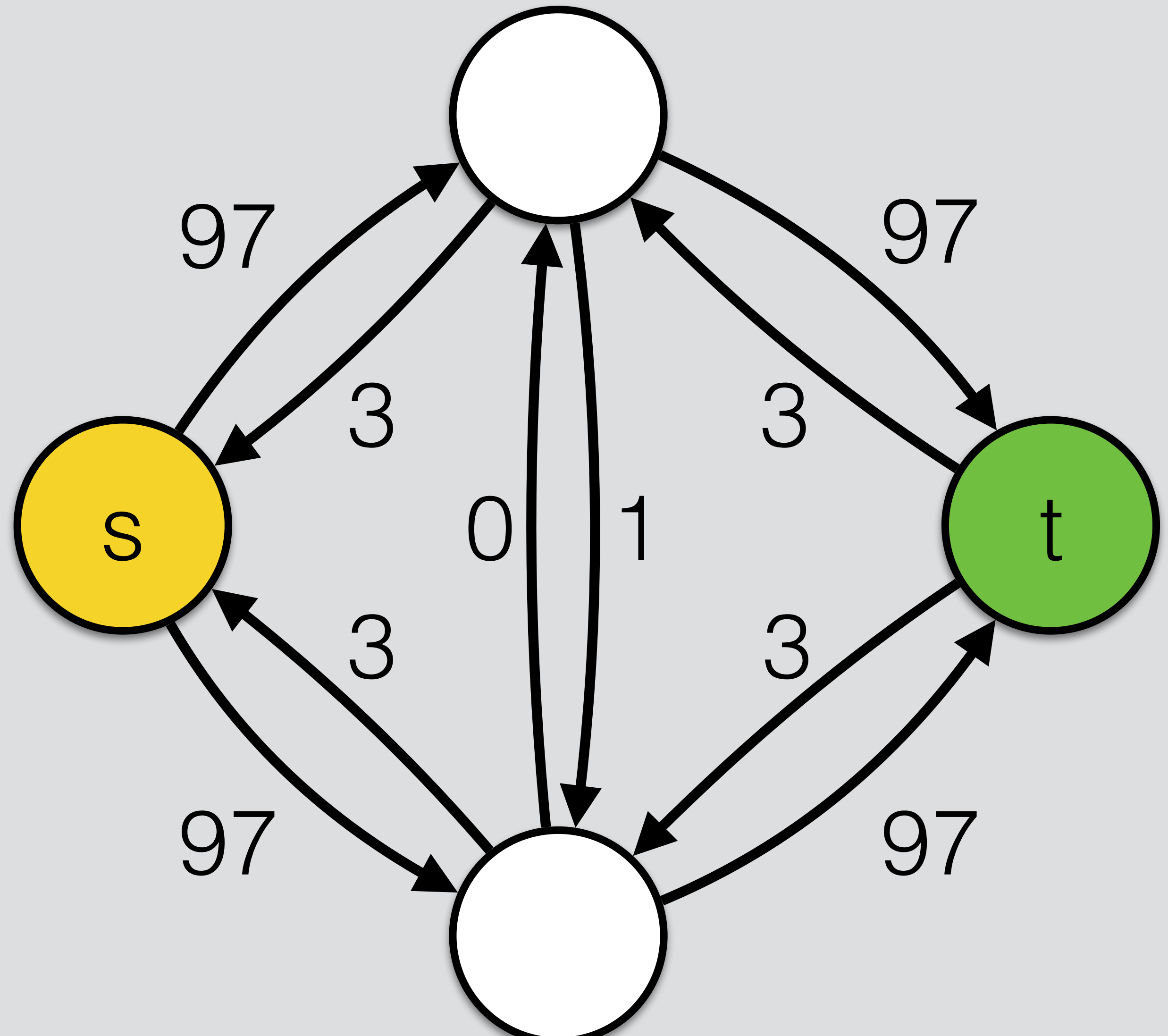
Residual graph



Graph (with flow)



Residual graph



Now repeat 194 more times...
for a total of 200 = max-flow iterations.

Modifying Ford-Fulkerson (Edmonds-Karp)

Ford-Fulkerson Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`)

Create an empty map `F`.

for (every edge (v_i, v_j) in E):

    if ((v_j, v_i) is not in `E`):

        Add (v_j, v_i) to `E` with capacity 0.

    $F[(v_i, v_j)] = 0;$

while (true):

Can be *any* augmenting path.

    Let p = an augmenting path in residual of G.

    If no `p` exists, `break;`

    f_m = minimum weight of edge in `p`.

    Update flow along edges of `p` by f_m.

Edmonds-Karp Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`)

Create an empty map `F`.

for (every edge (v_i, v_j) in E):

    if ((v_j, v_i) is not in `E`):

        Add (v_j, v_i) to `E` with capacity 0.

    $F[(v_i, v_j)] = 0;$

while (true):

Find via BFS

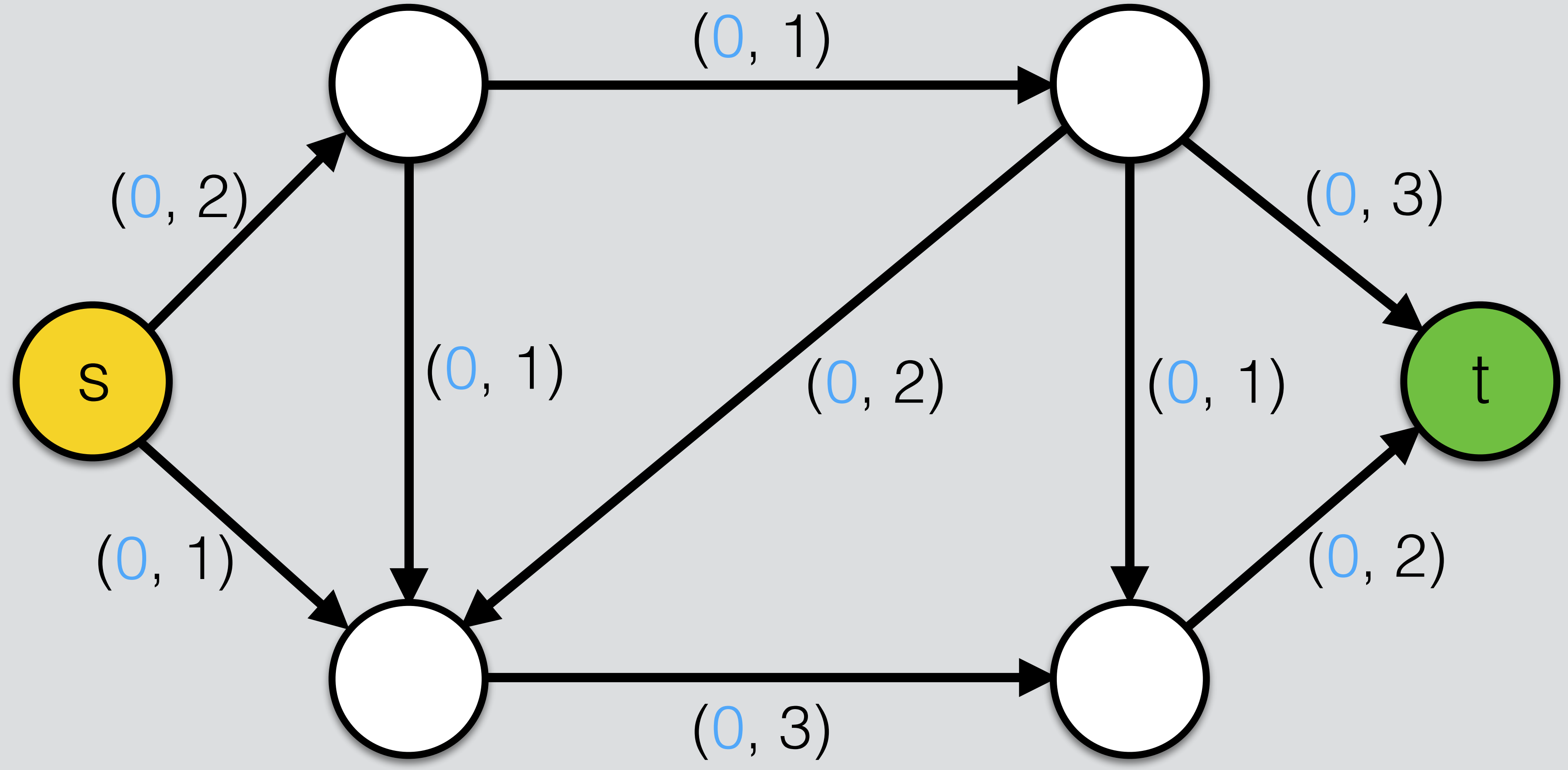
    Let p = an (edge-length-)shortest augmenting path in residual of G.

    If no `p` exists, `break;`

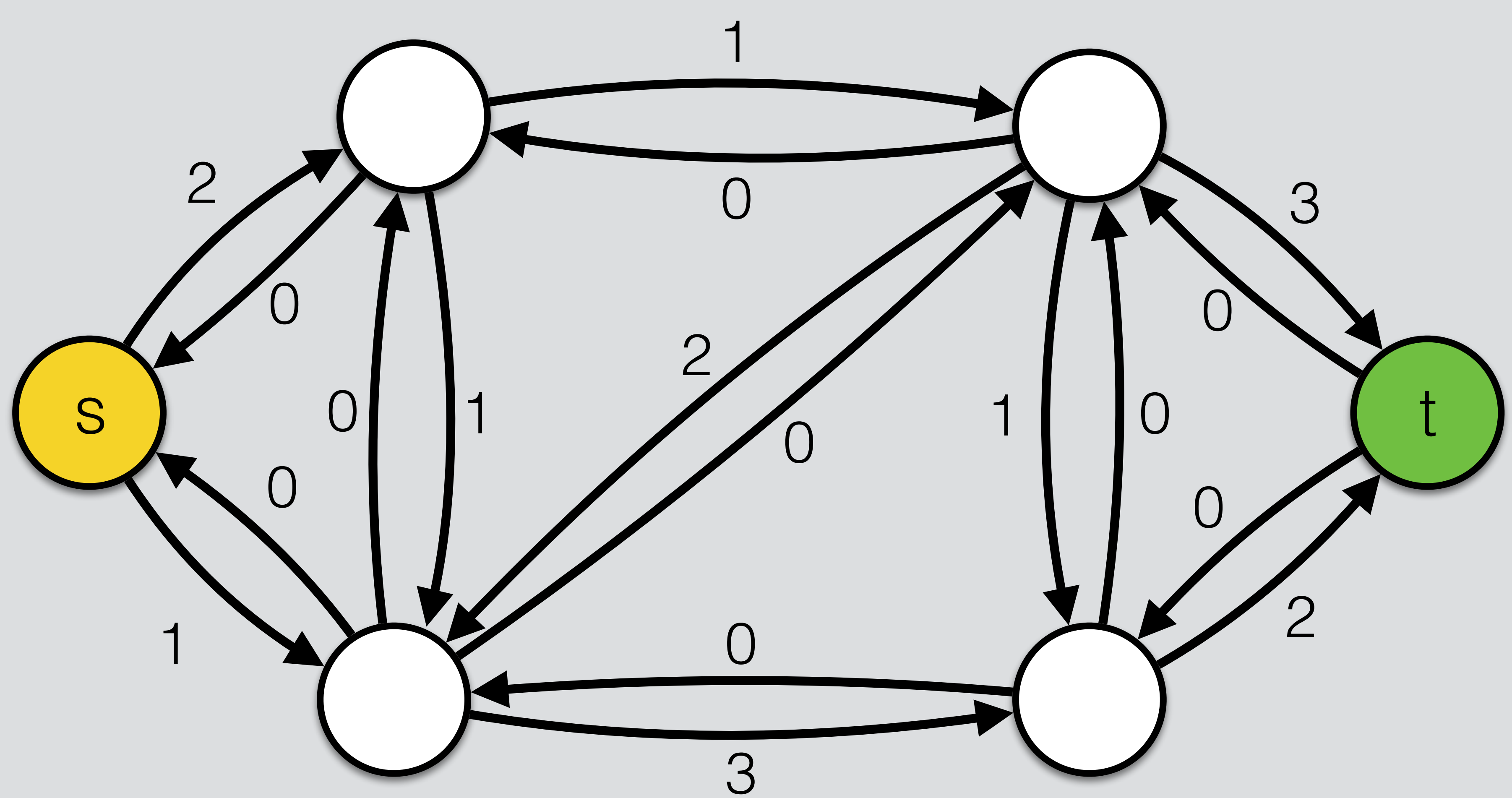
    f_m = minimum weight of edge in `p`.

    Update flow along edges of `p` by f_m.

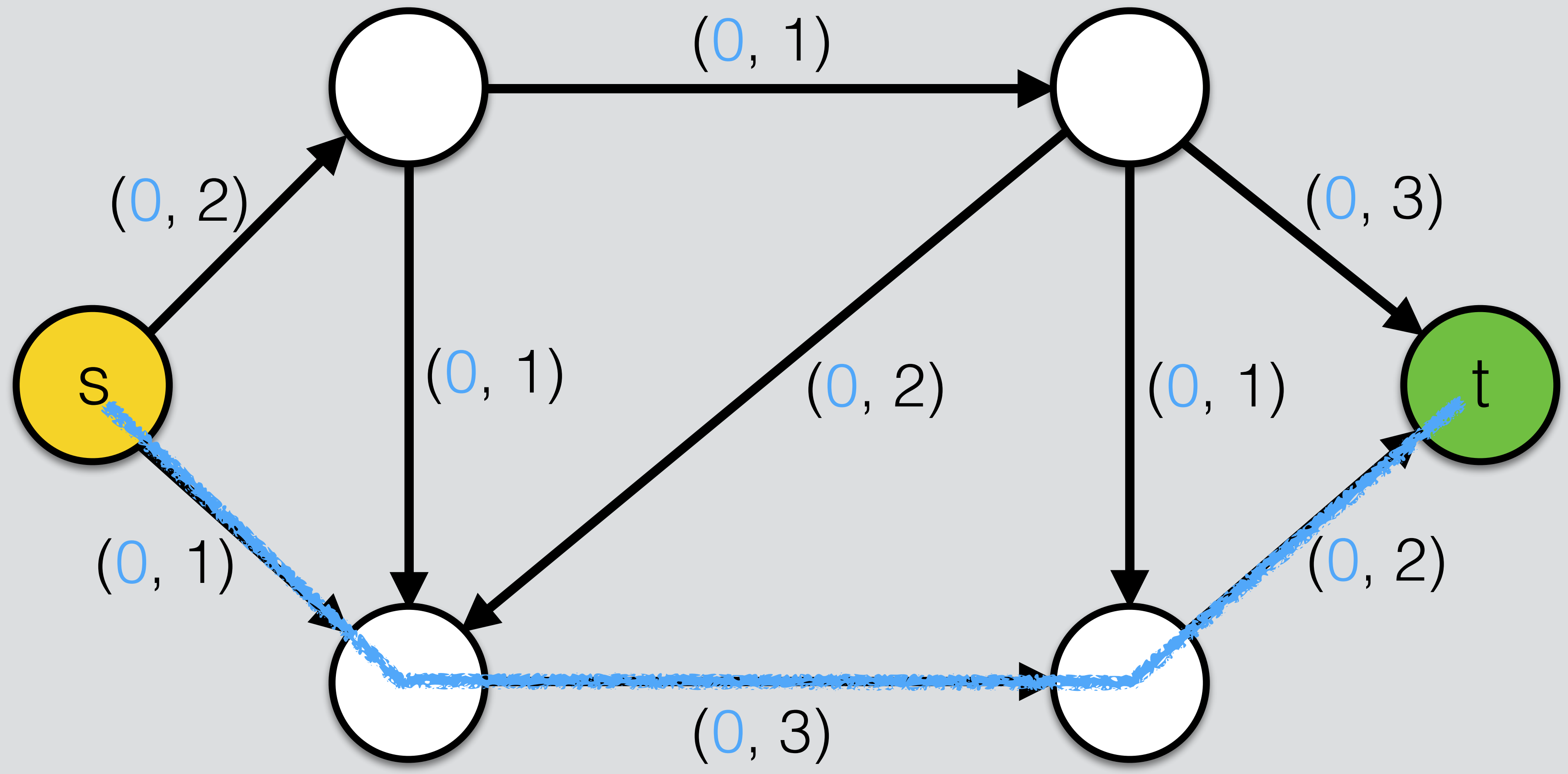
Input graph:



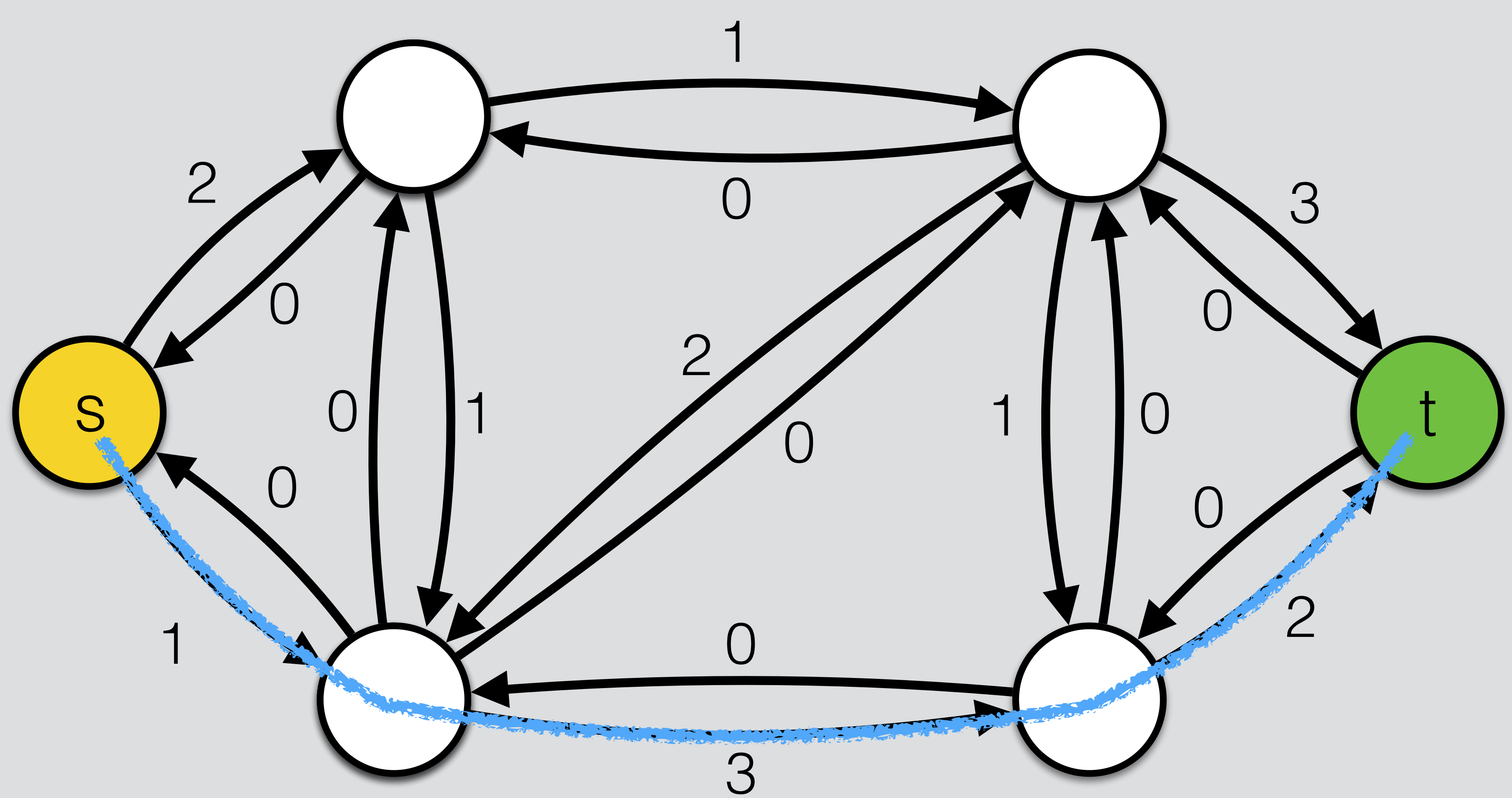
Residual graph:



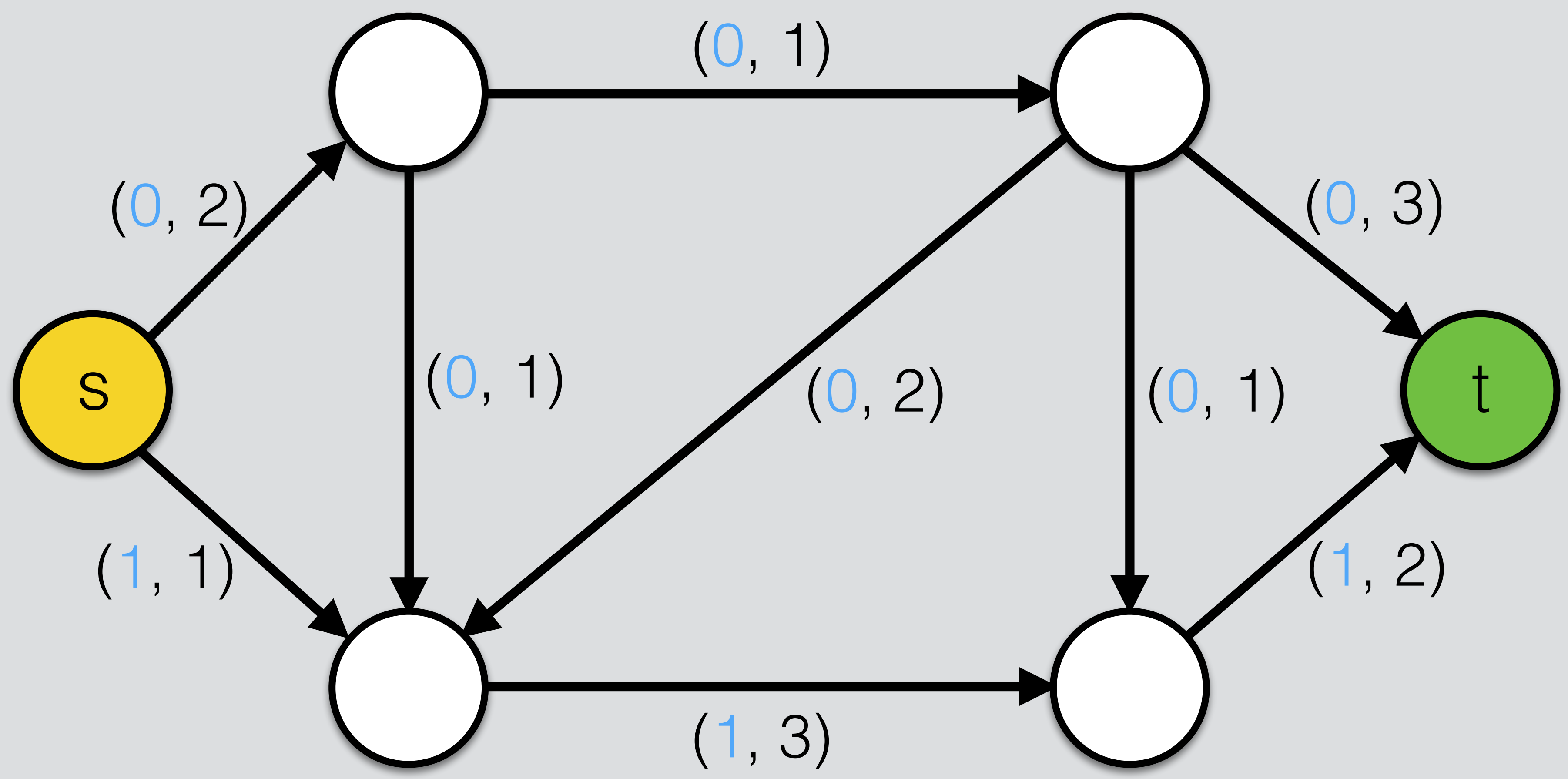
Input graph:



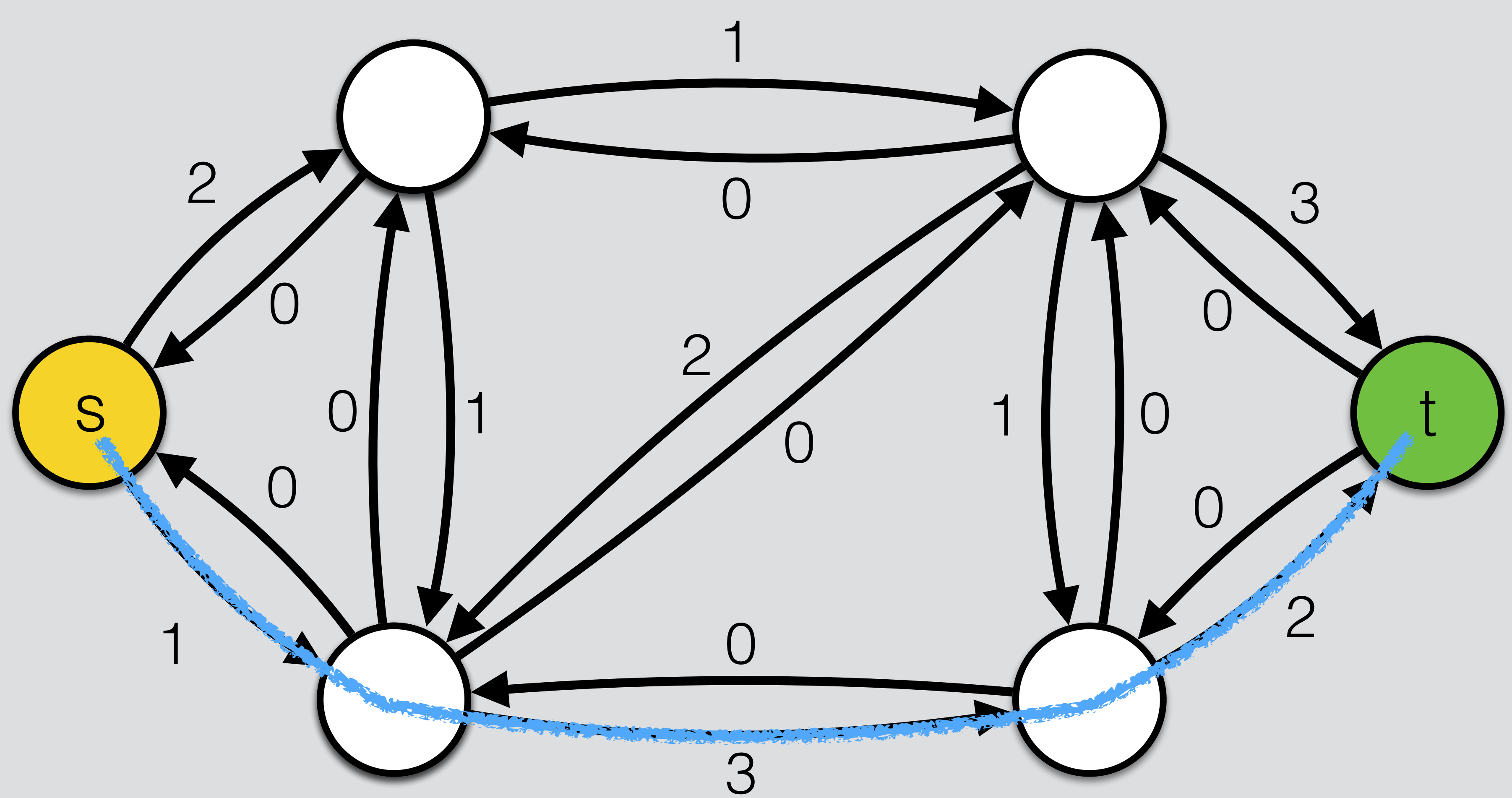
Residual graph:



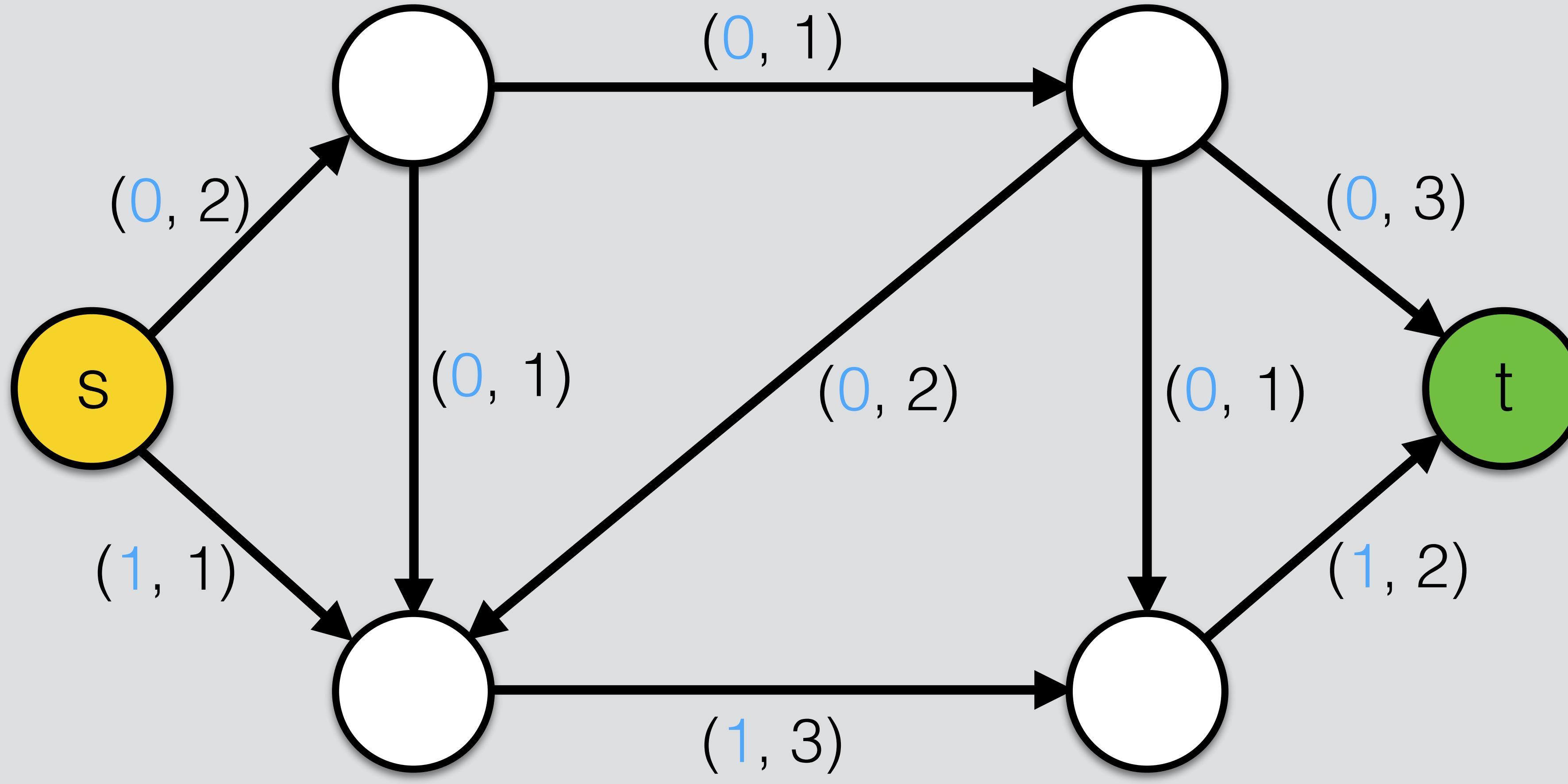
Input graph:



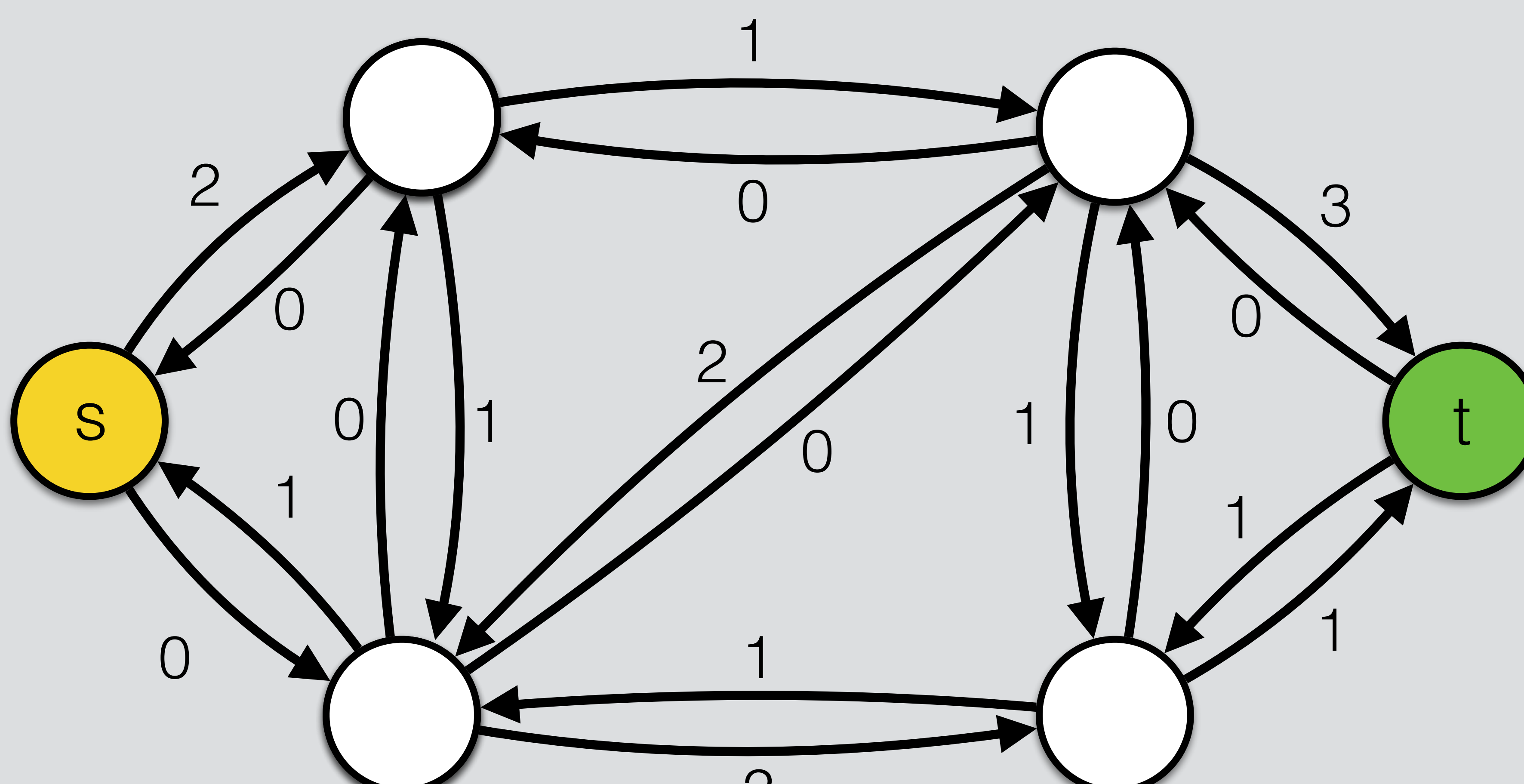
Residual graph:



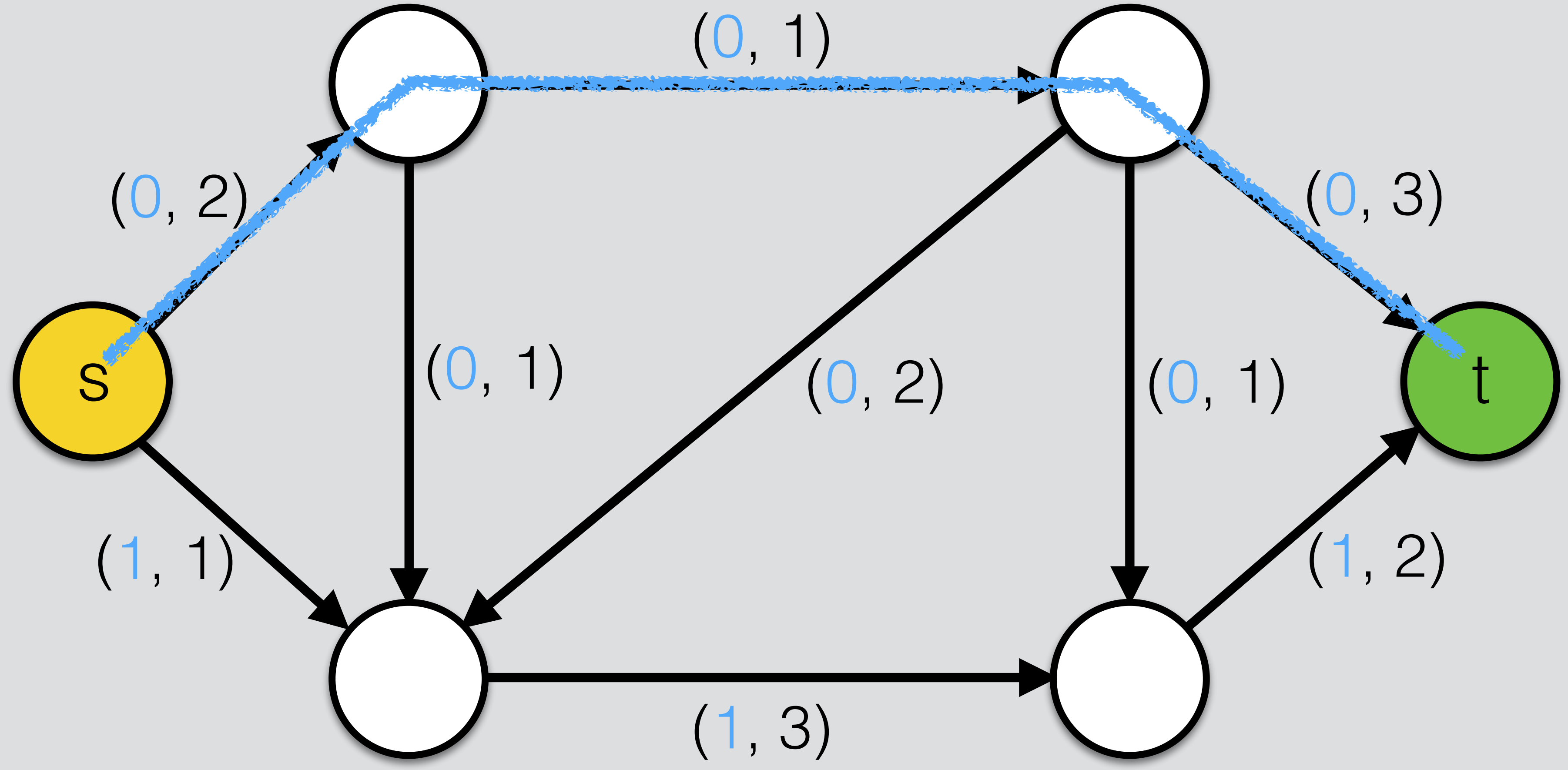
Input graph:



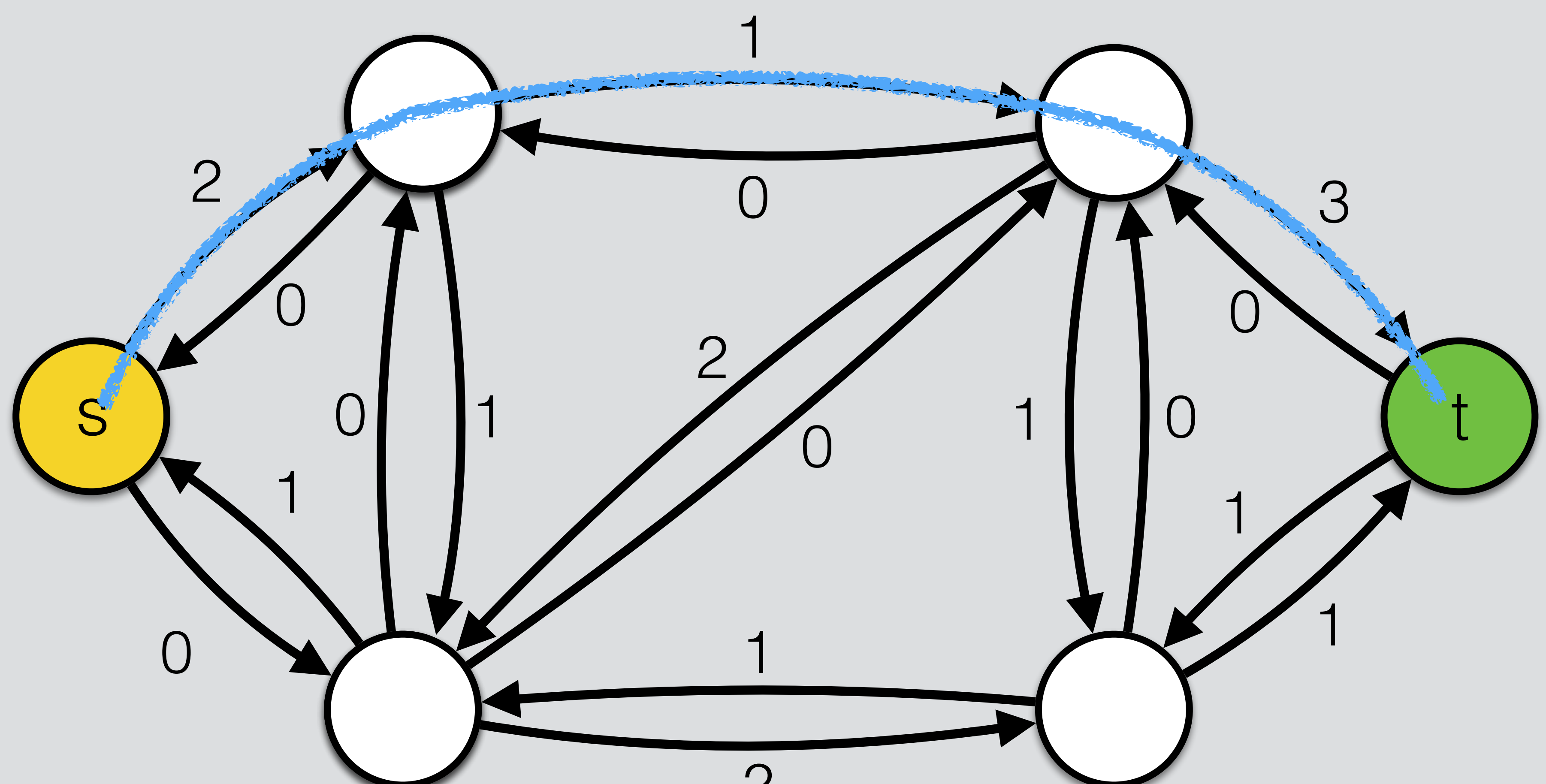
Residual graph:



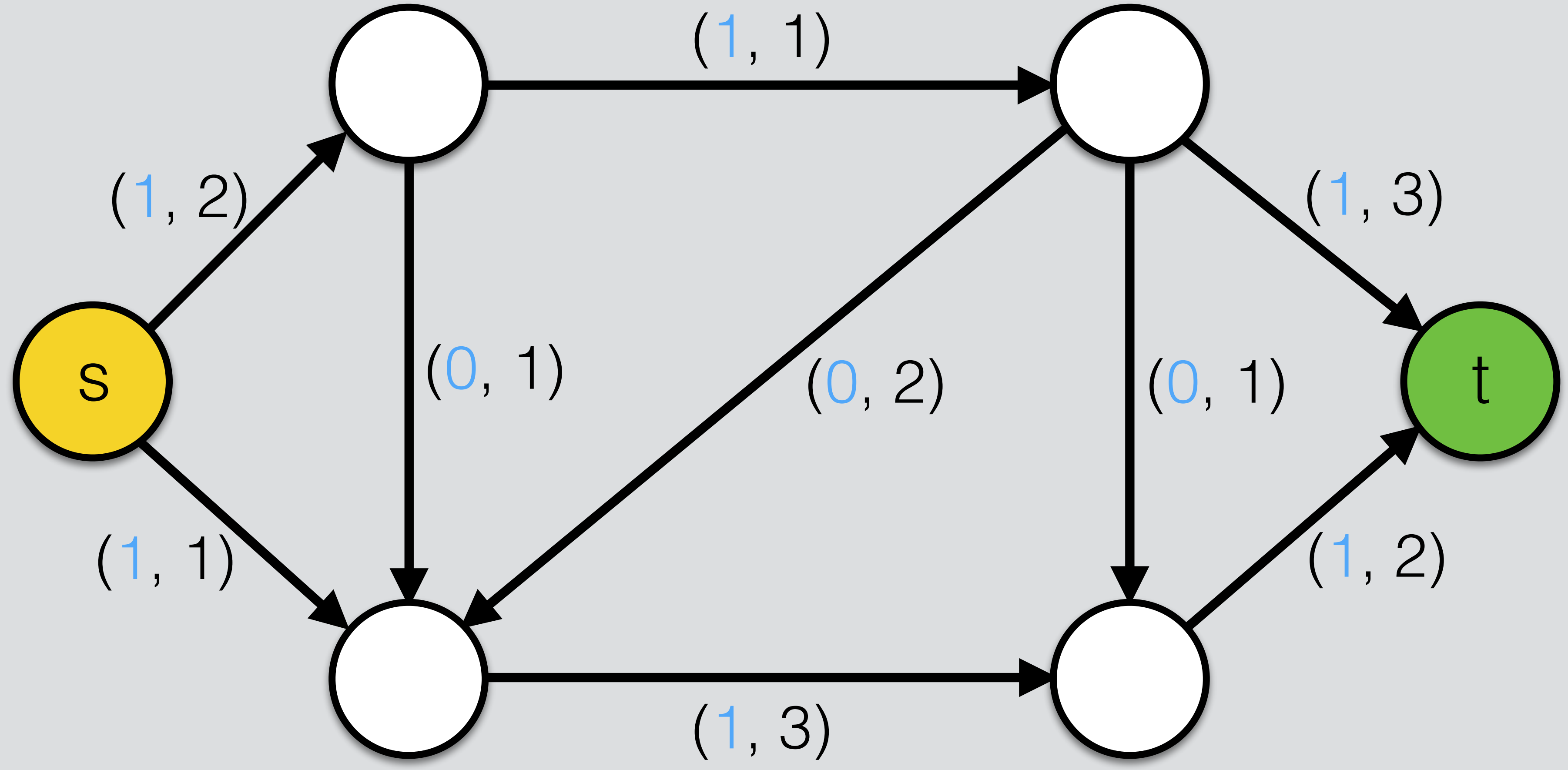
Input graph:



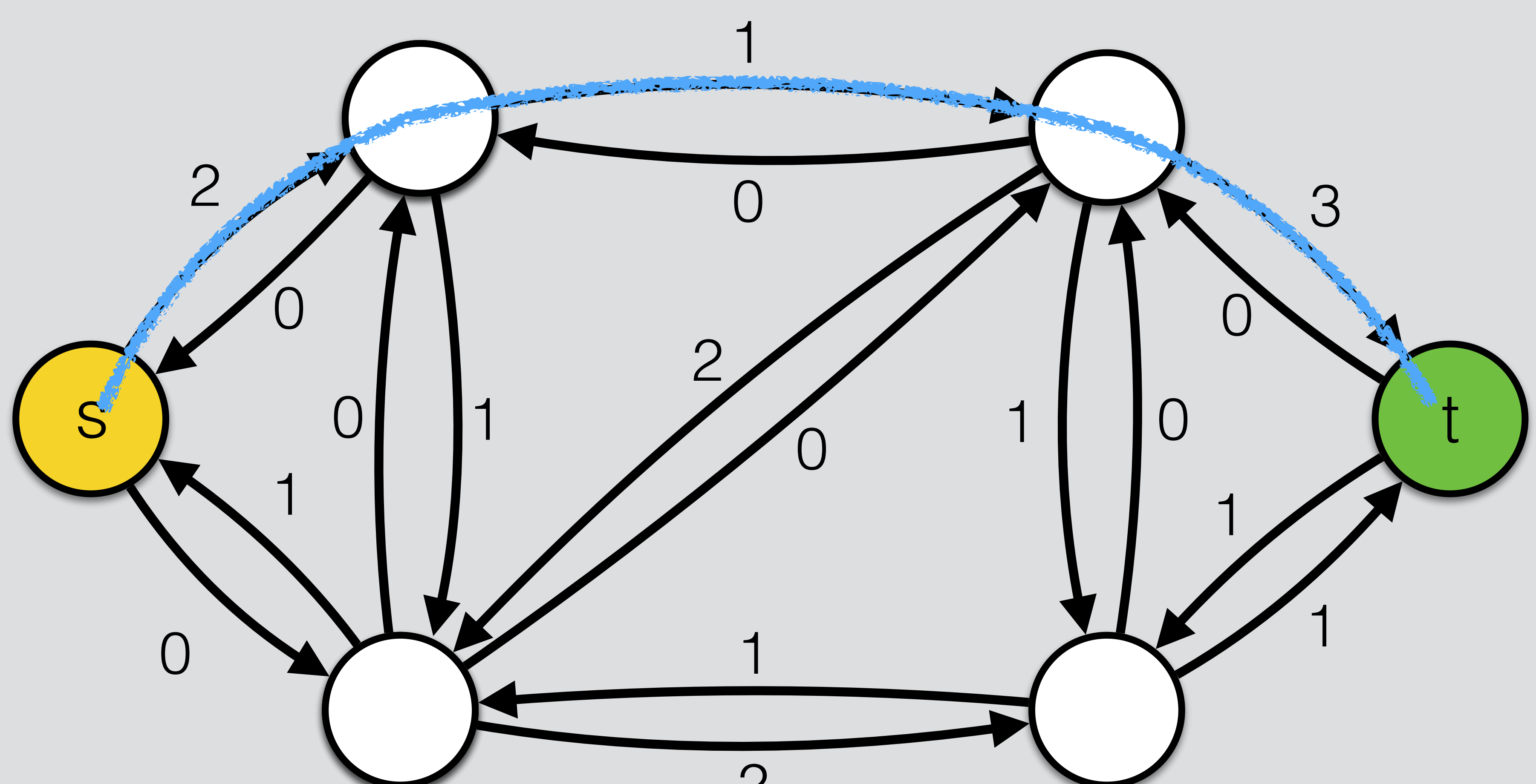
Residual graph:



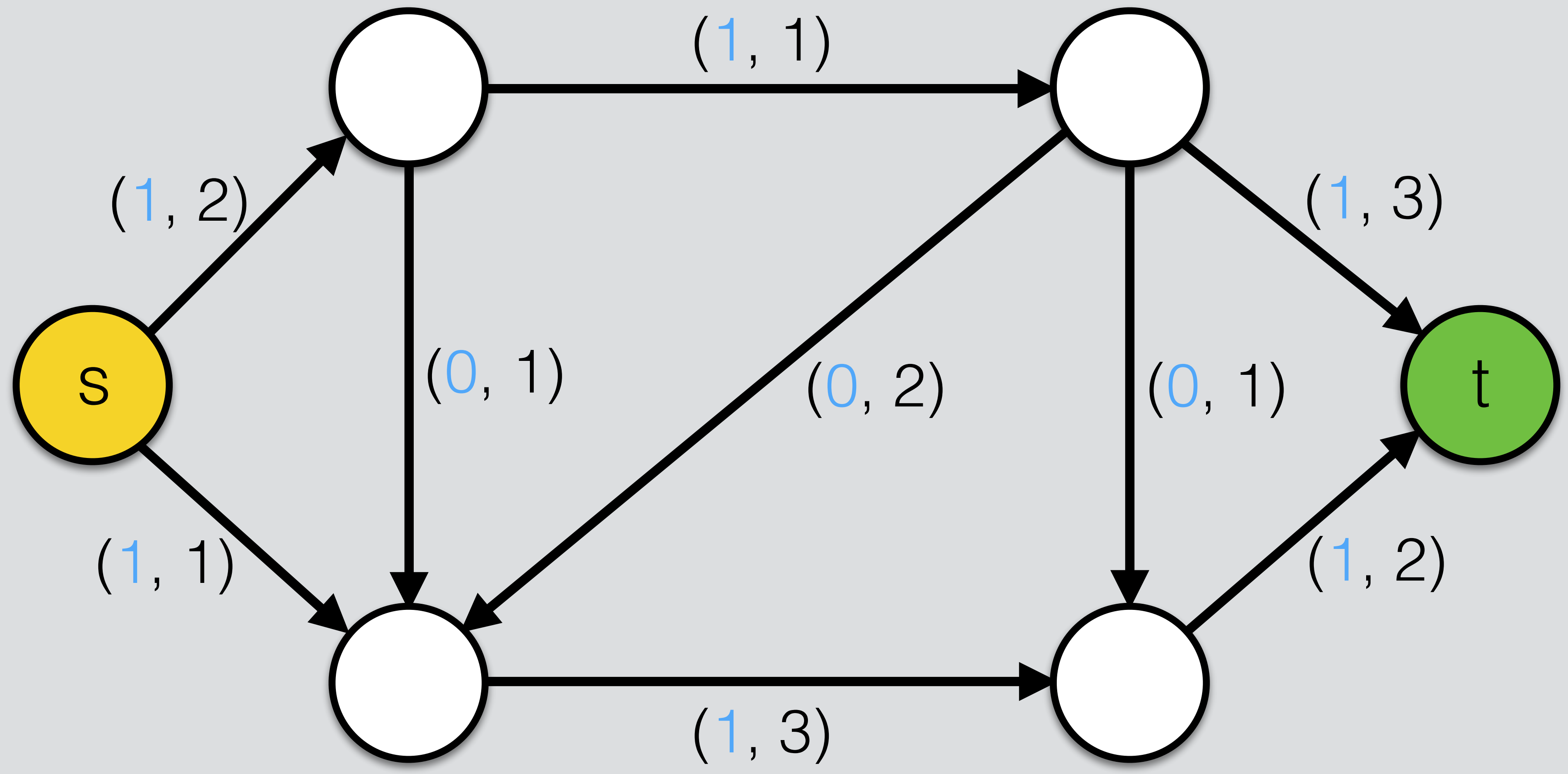
Input graph:



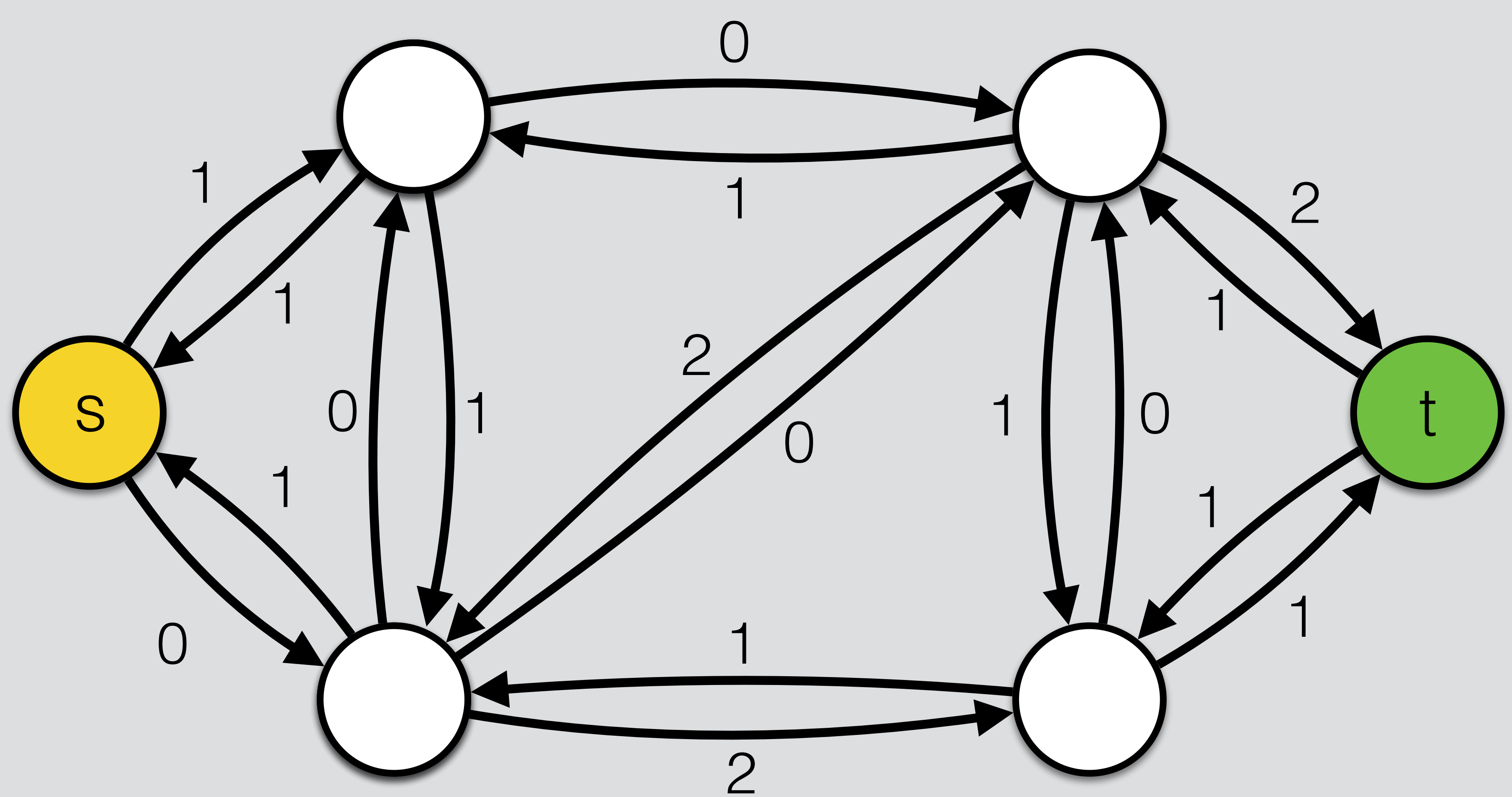
Residual graph:



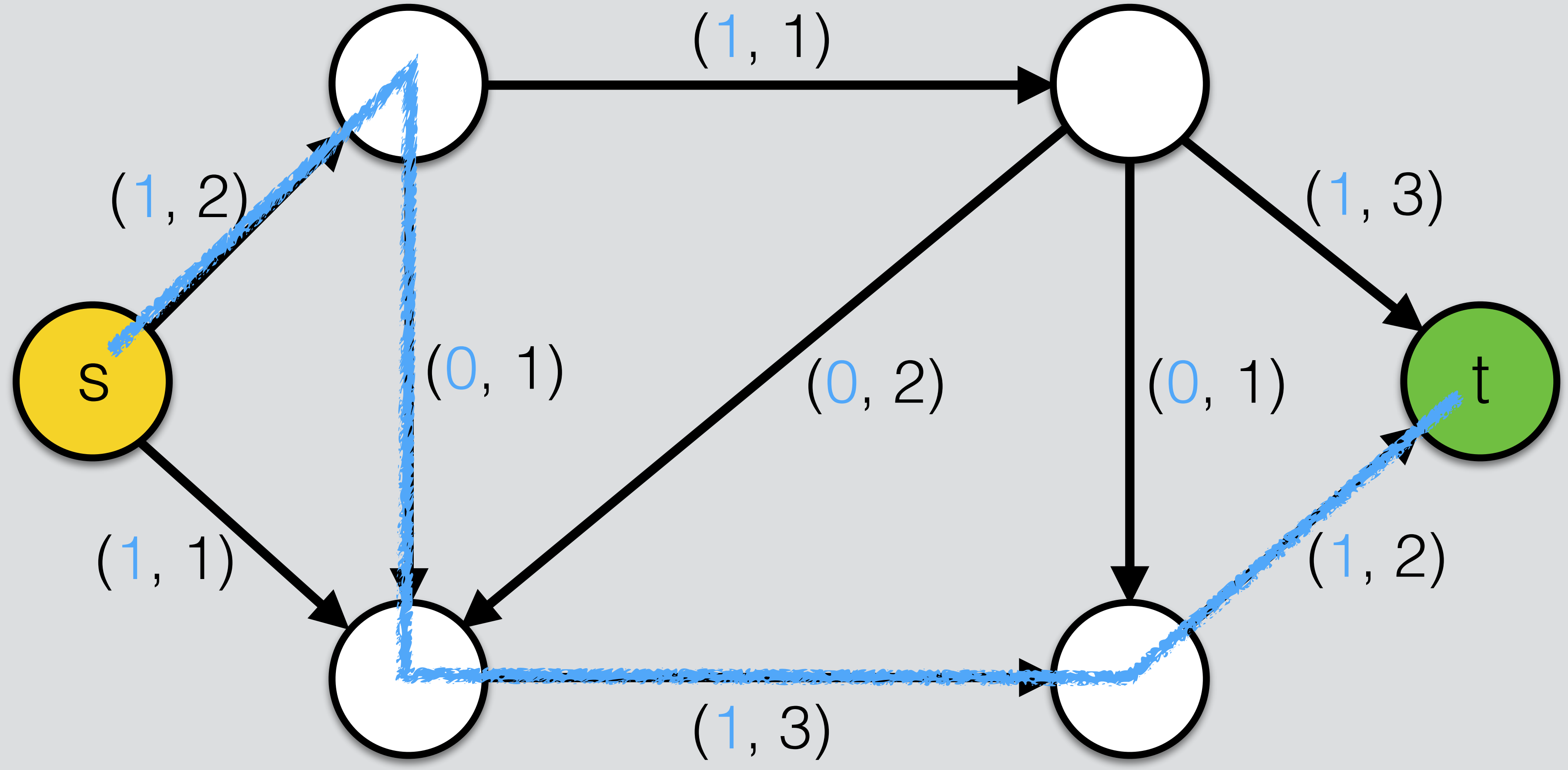
Input graph:



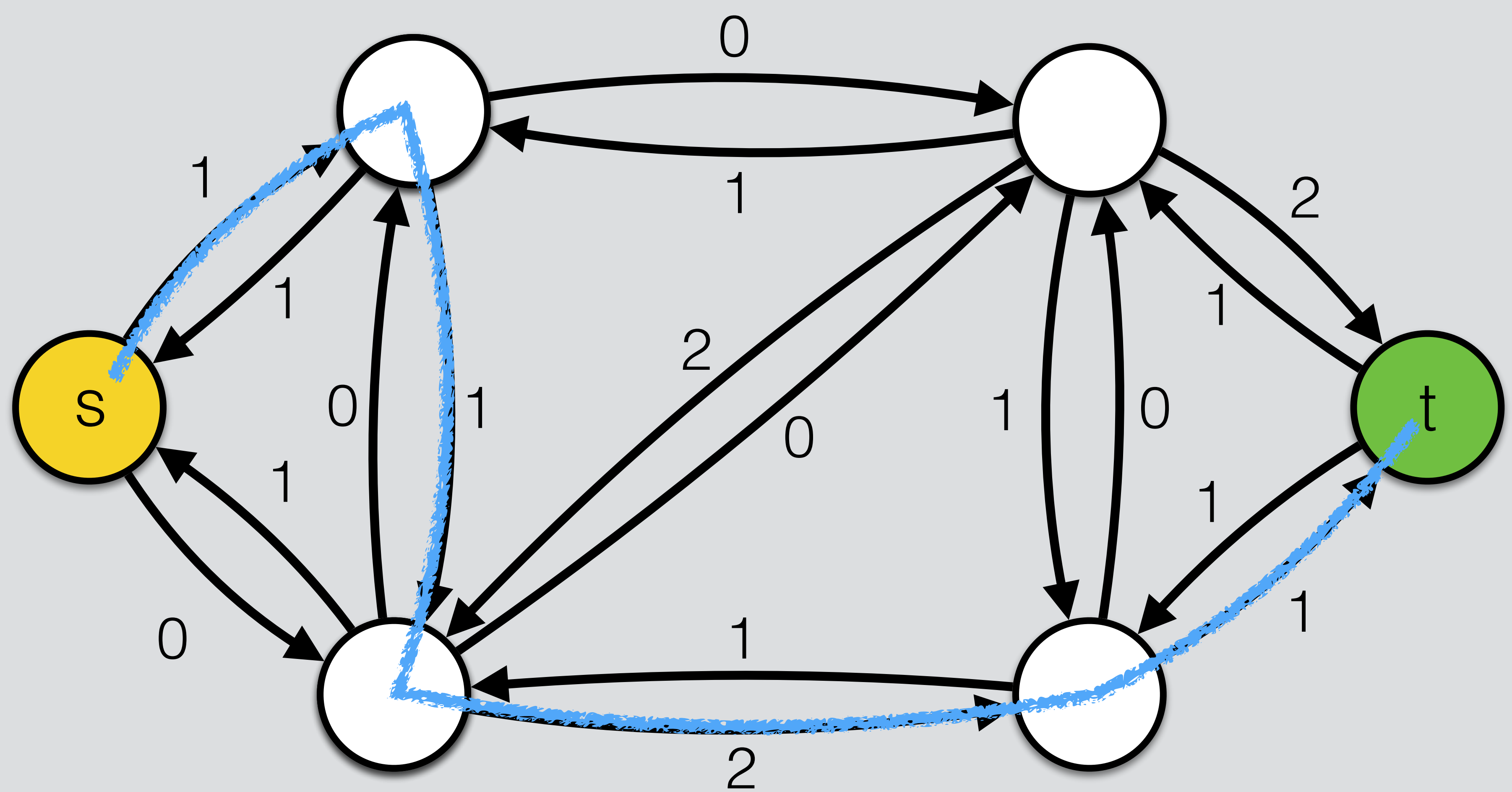
Residual graph:



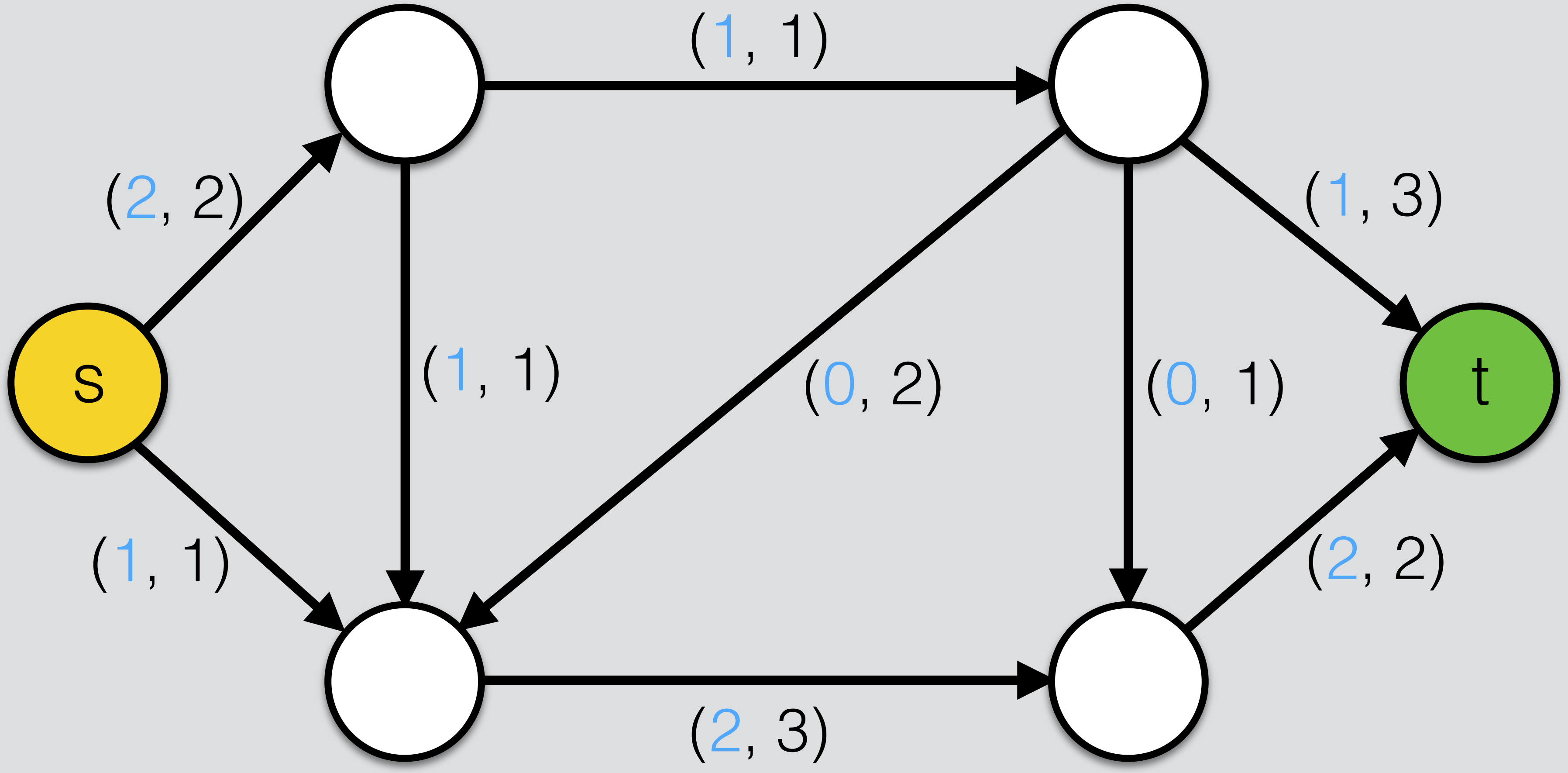
Input graph:



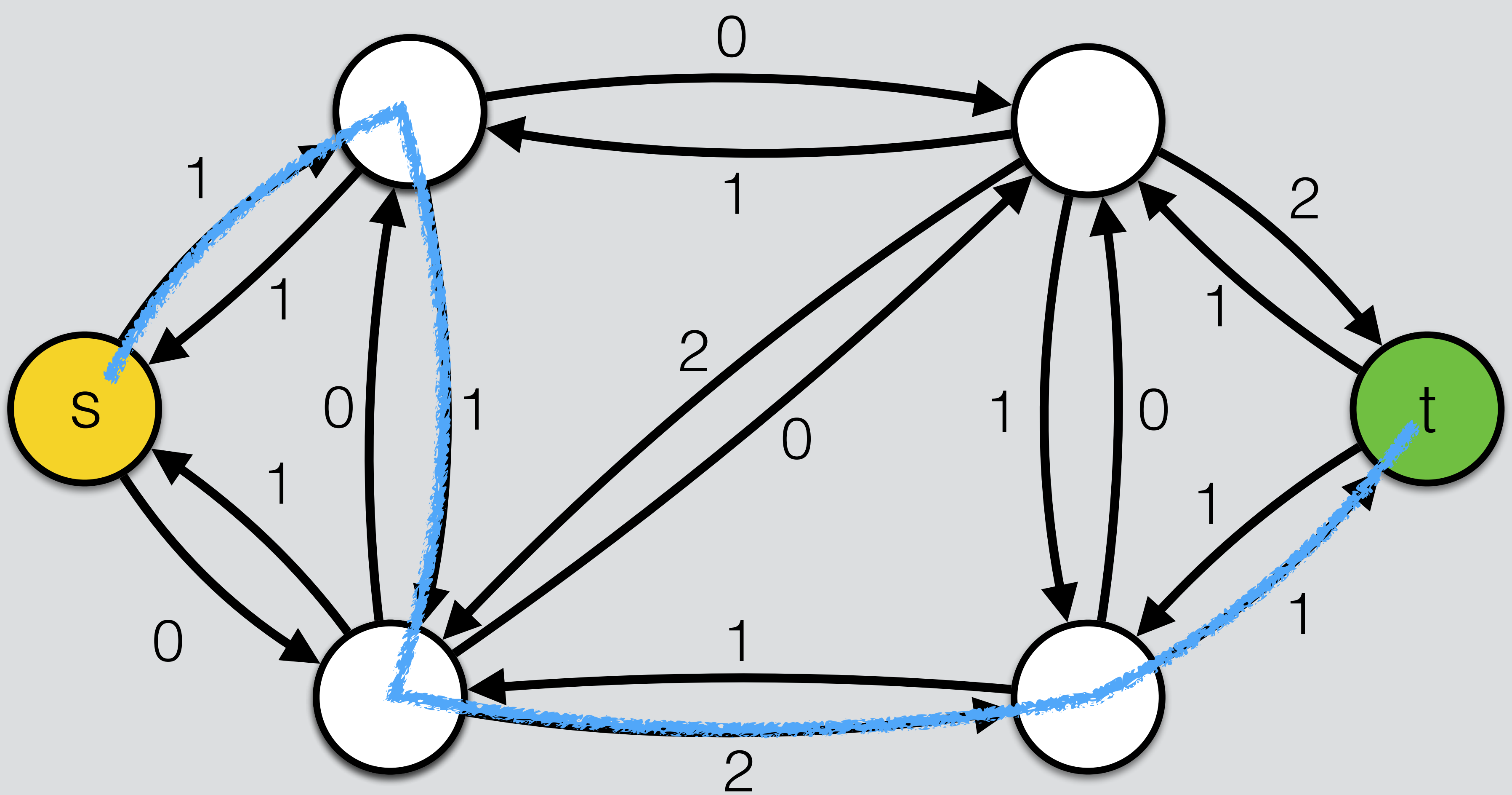
Residual graph:



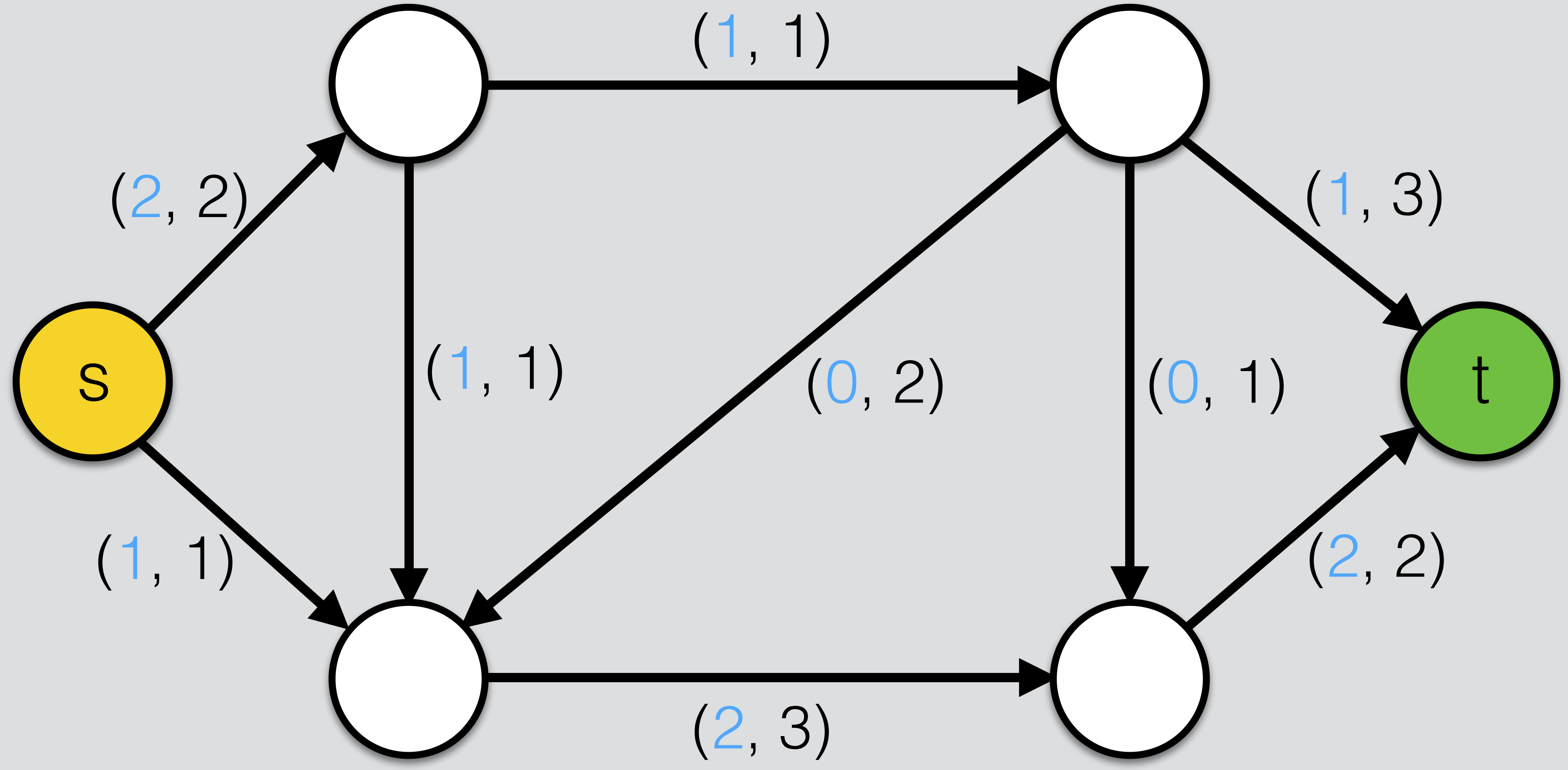
Input graph:



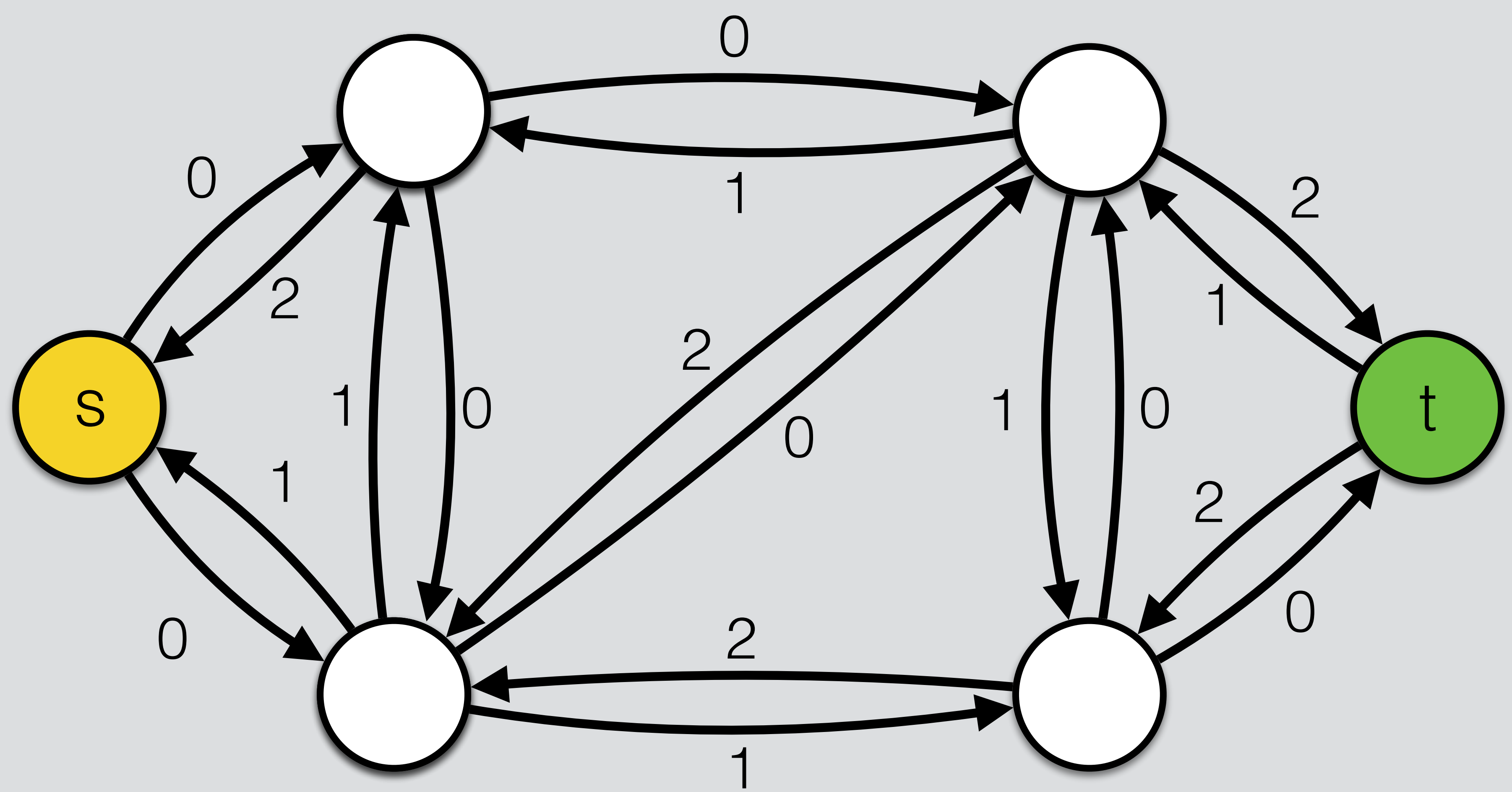
Residual graph:



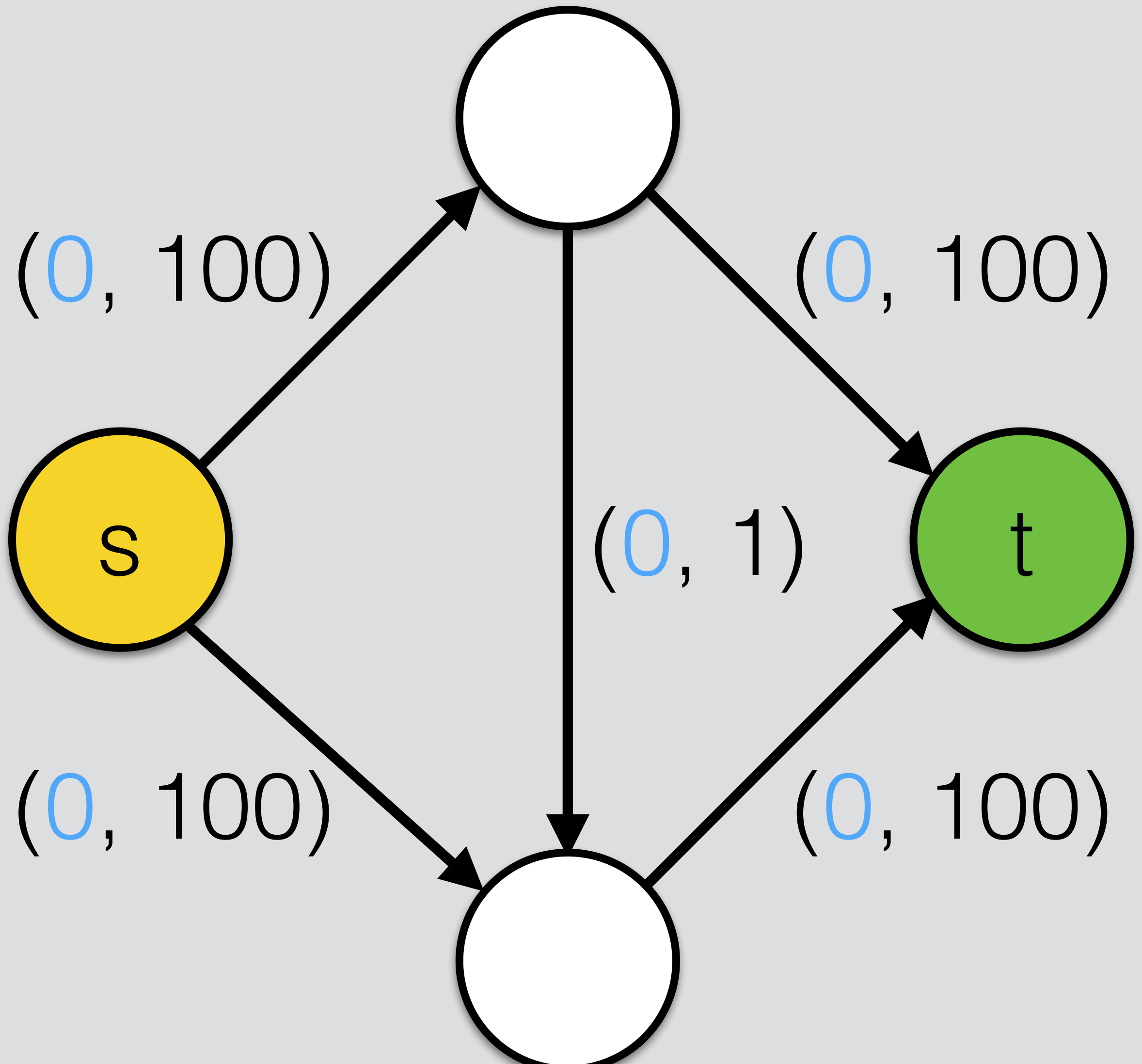
Input graph:



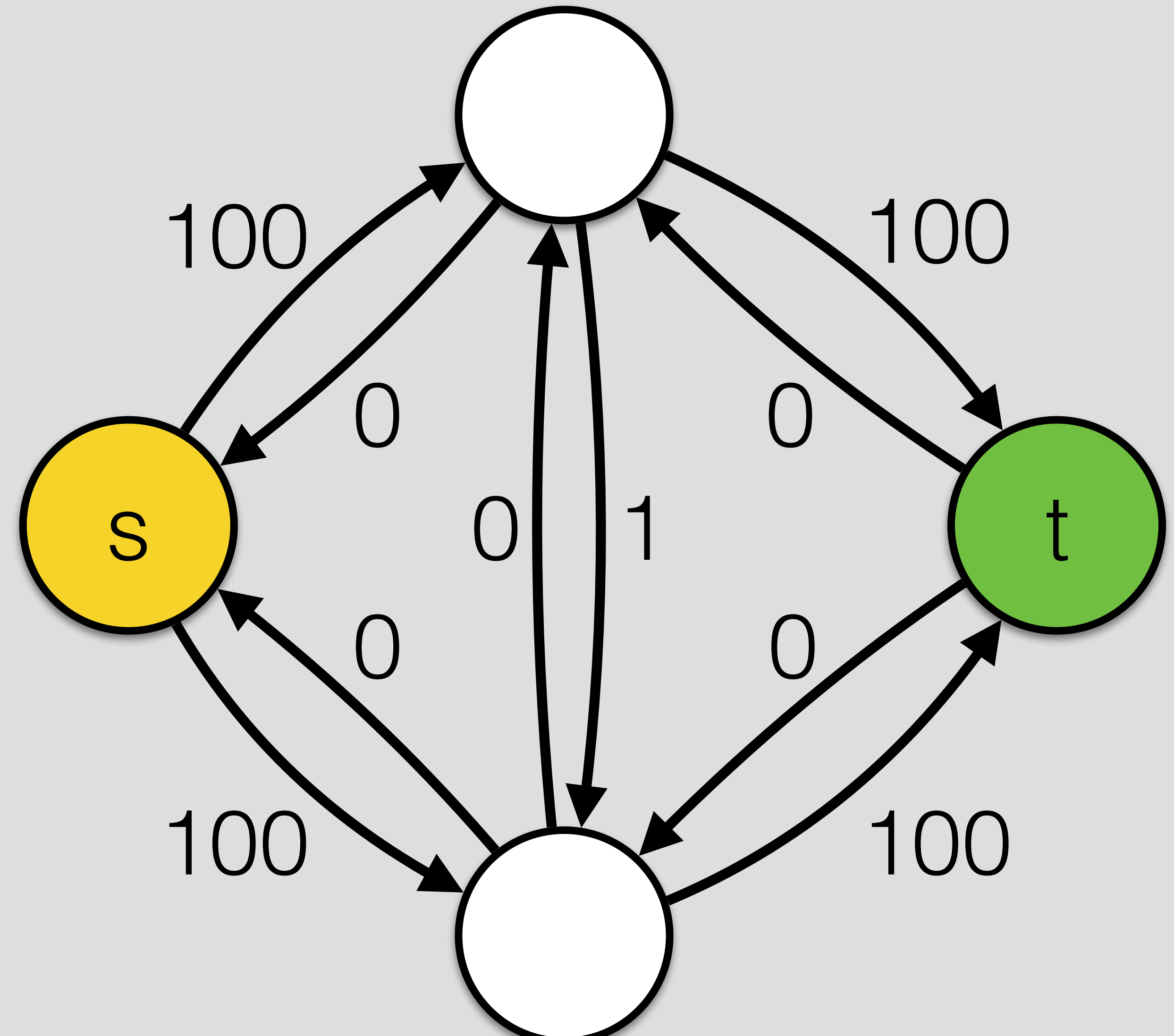
Residual graph:



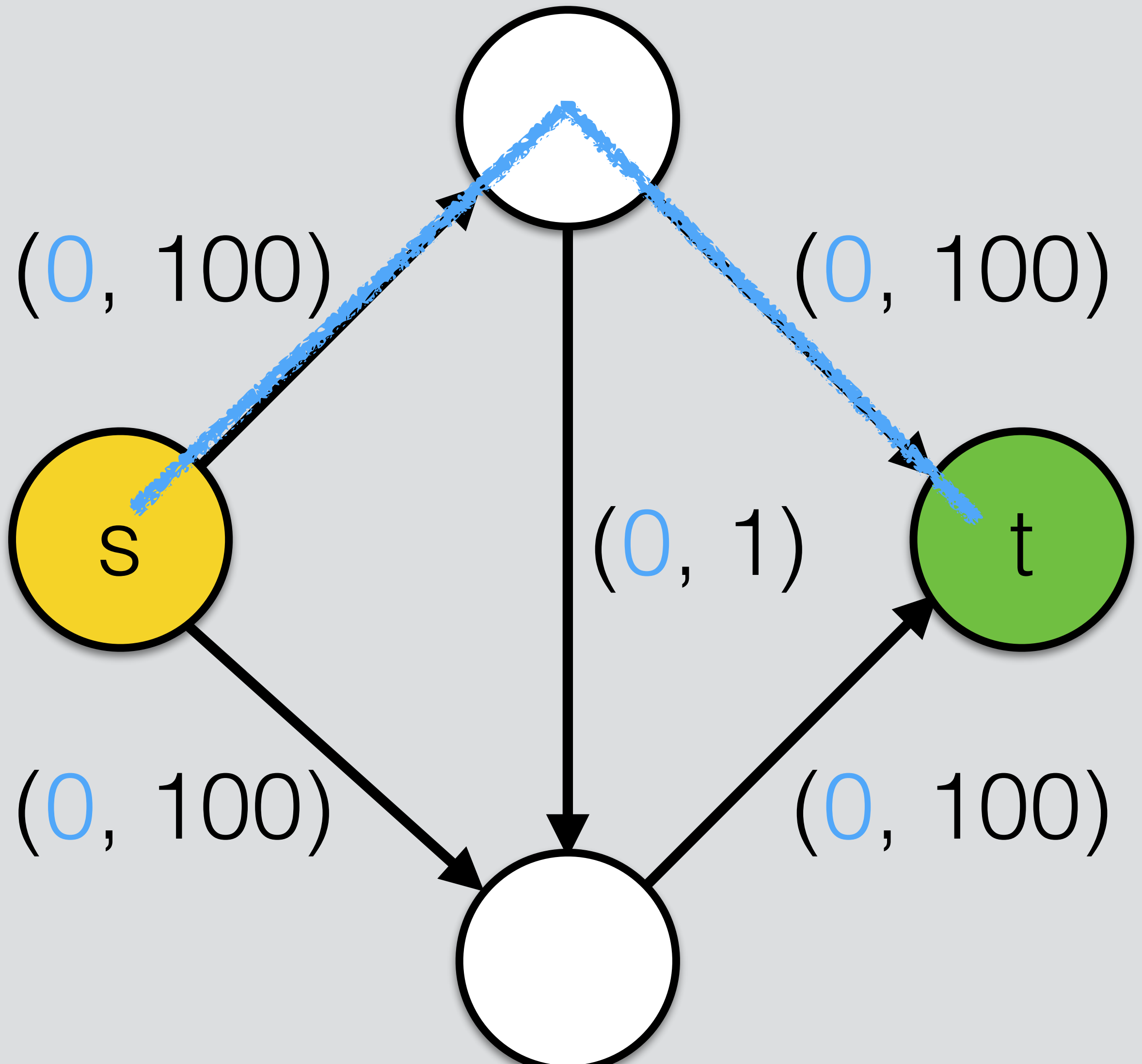
Graph (with flow)



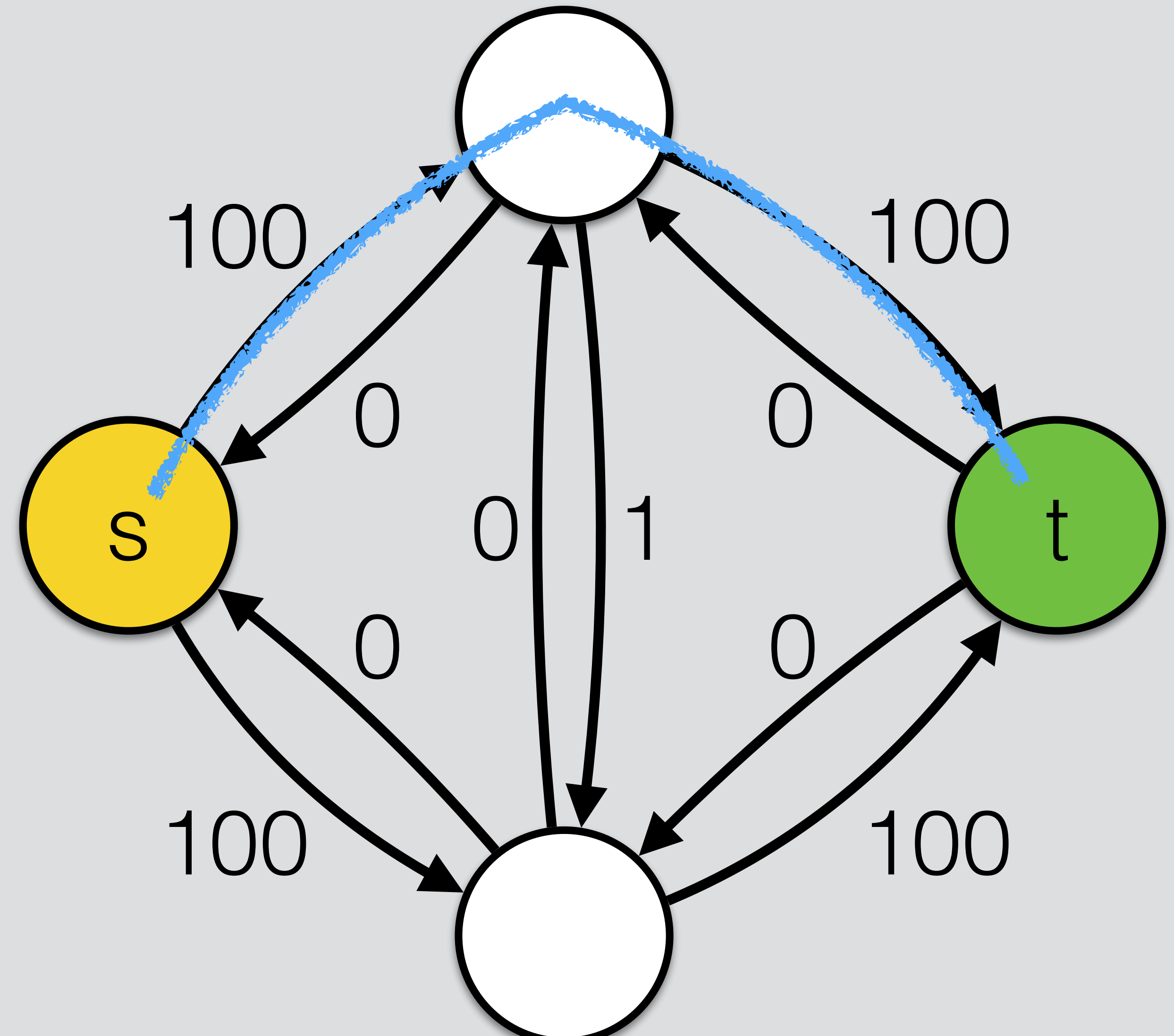
Residual graph



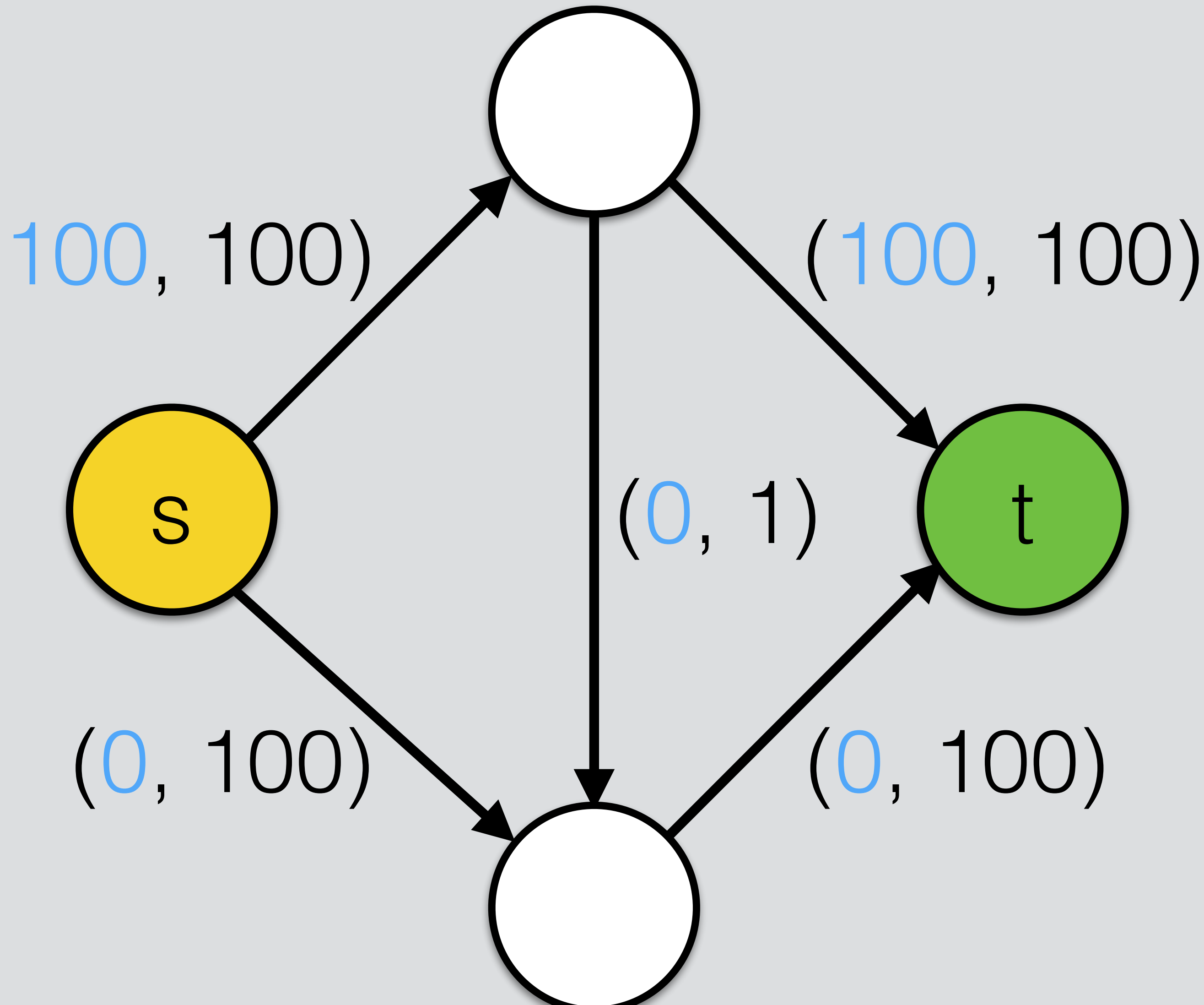
Graph (with flow)



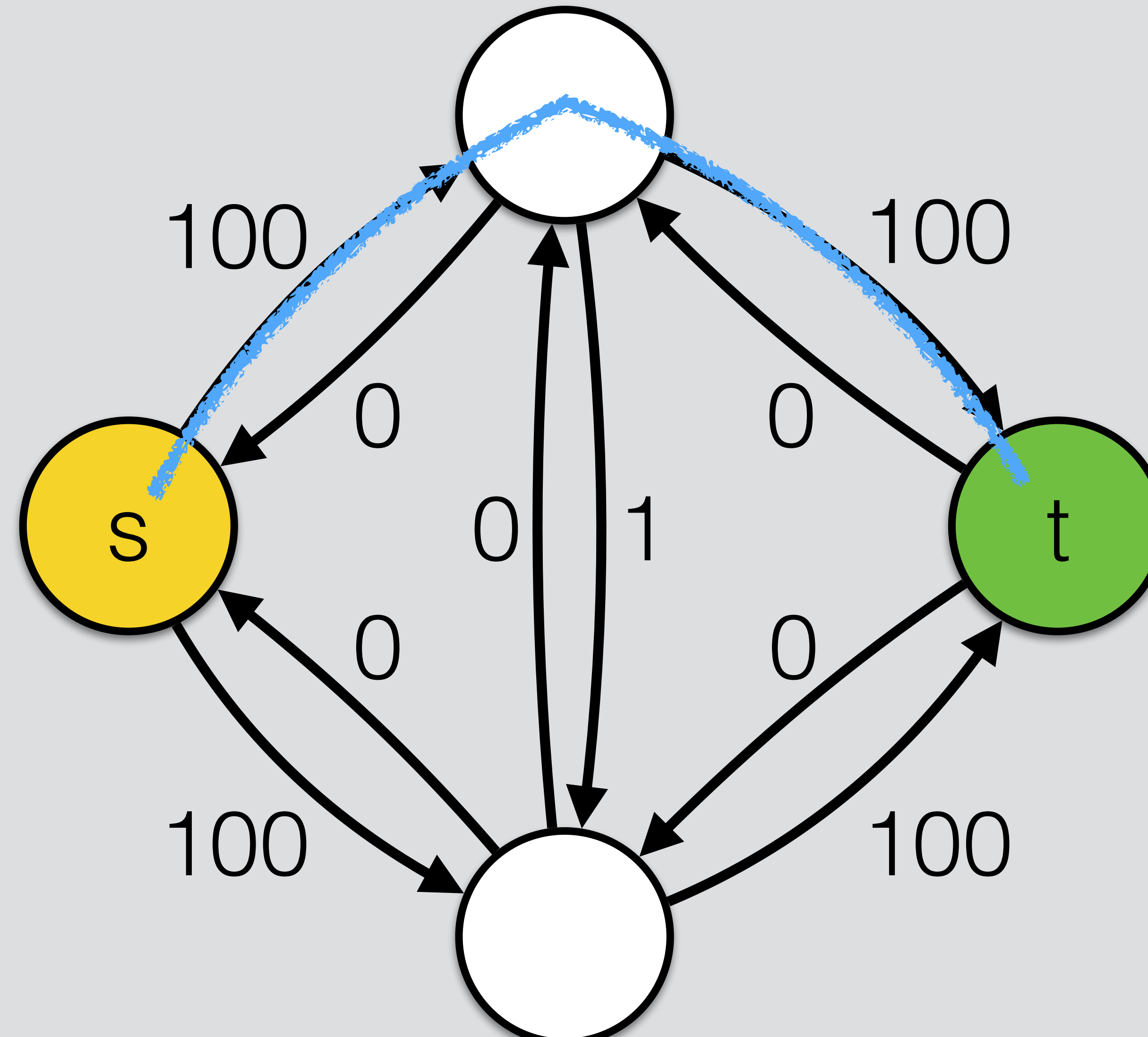
Residual graph



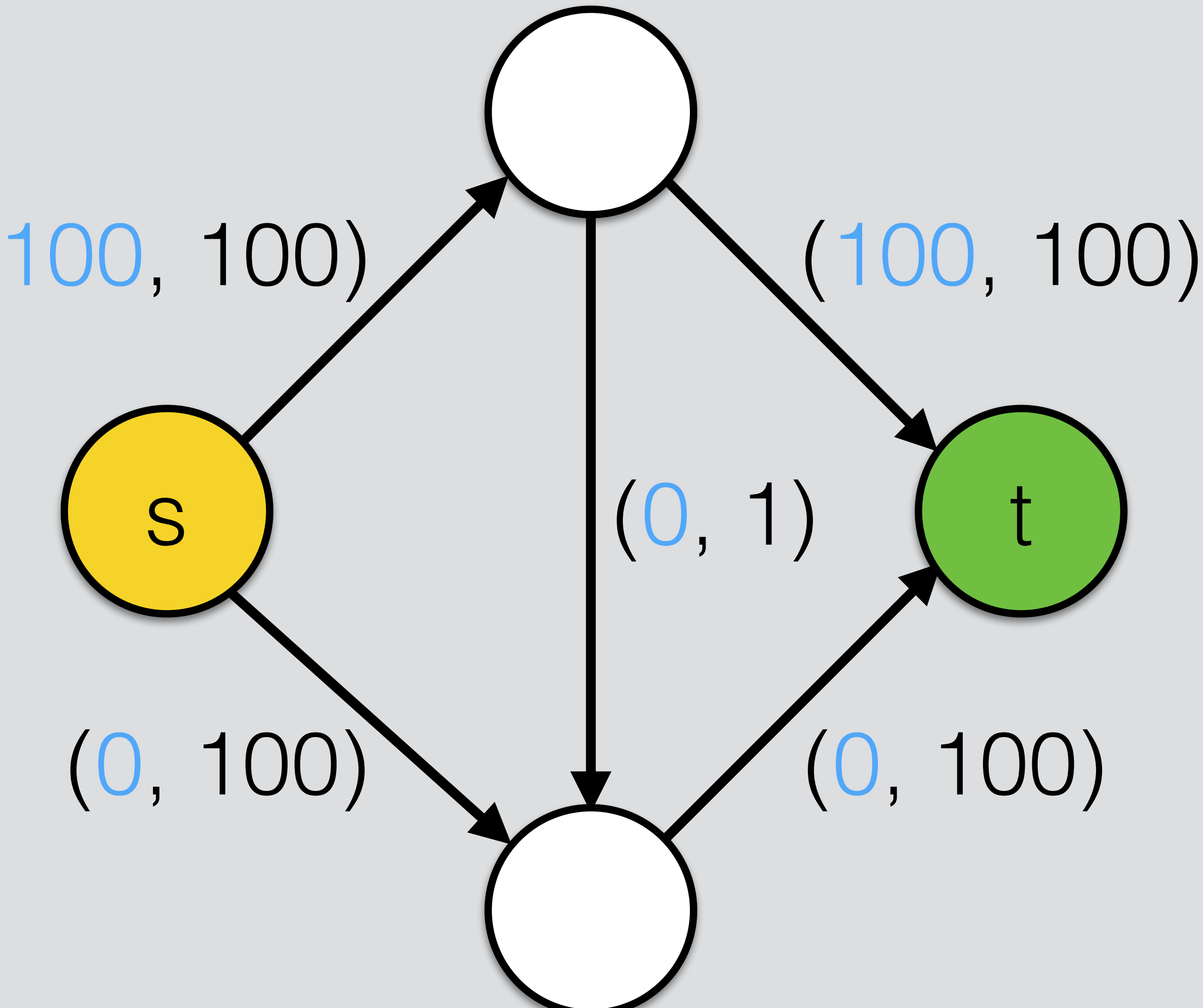
Graph (with flow)



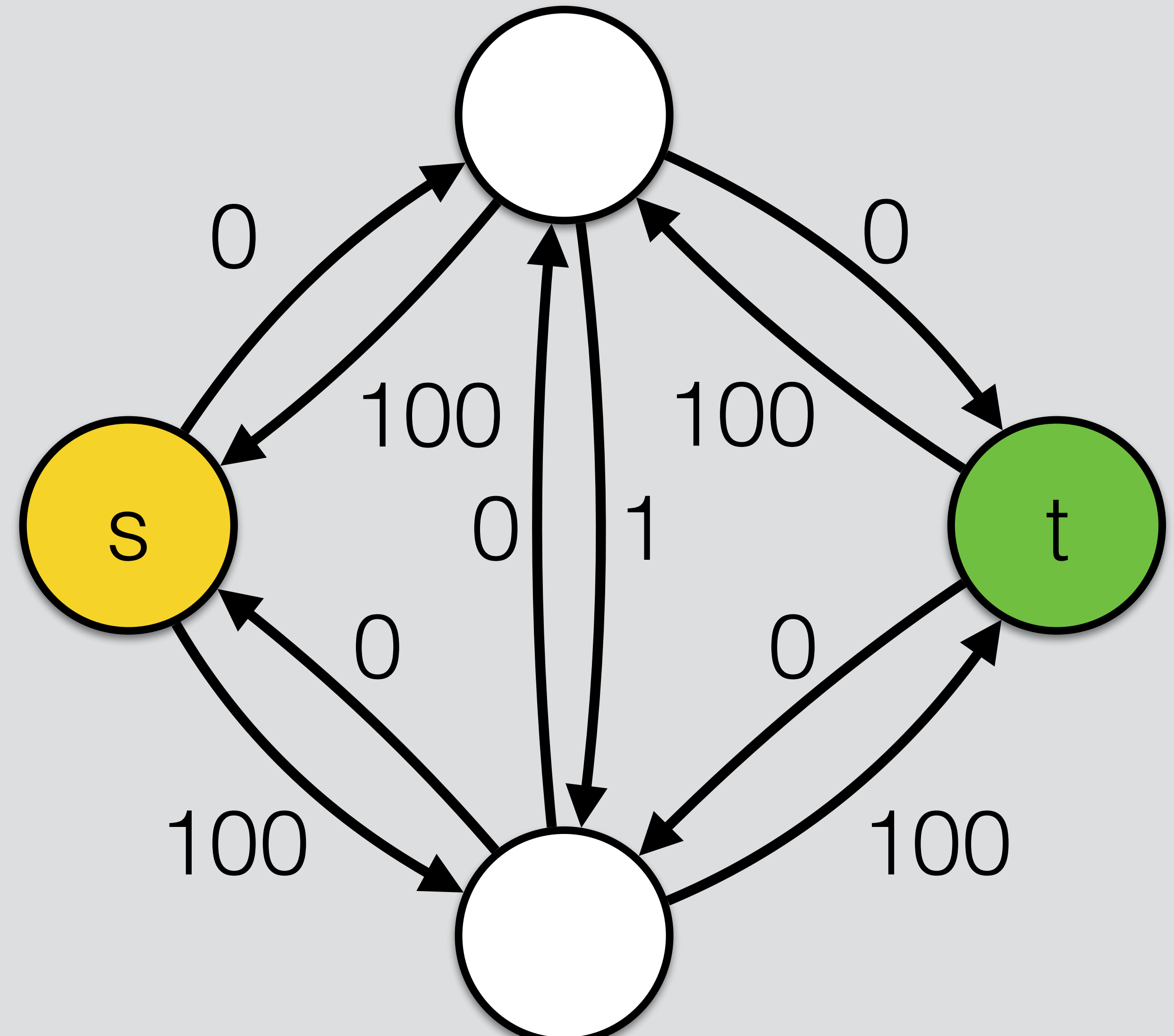
Residual graph



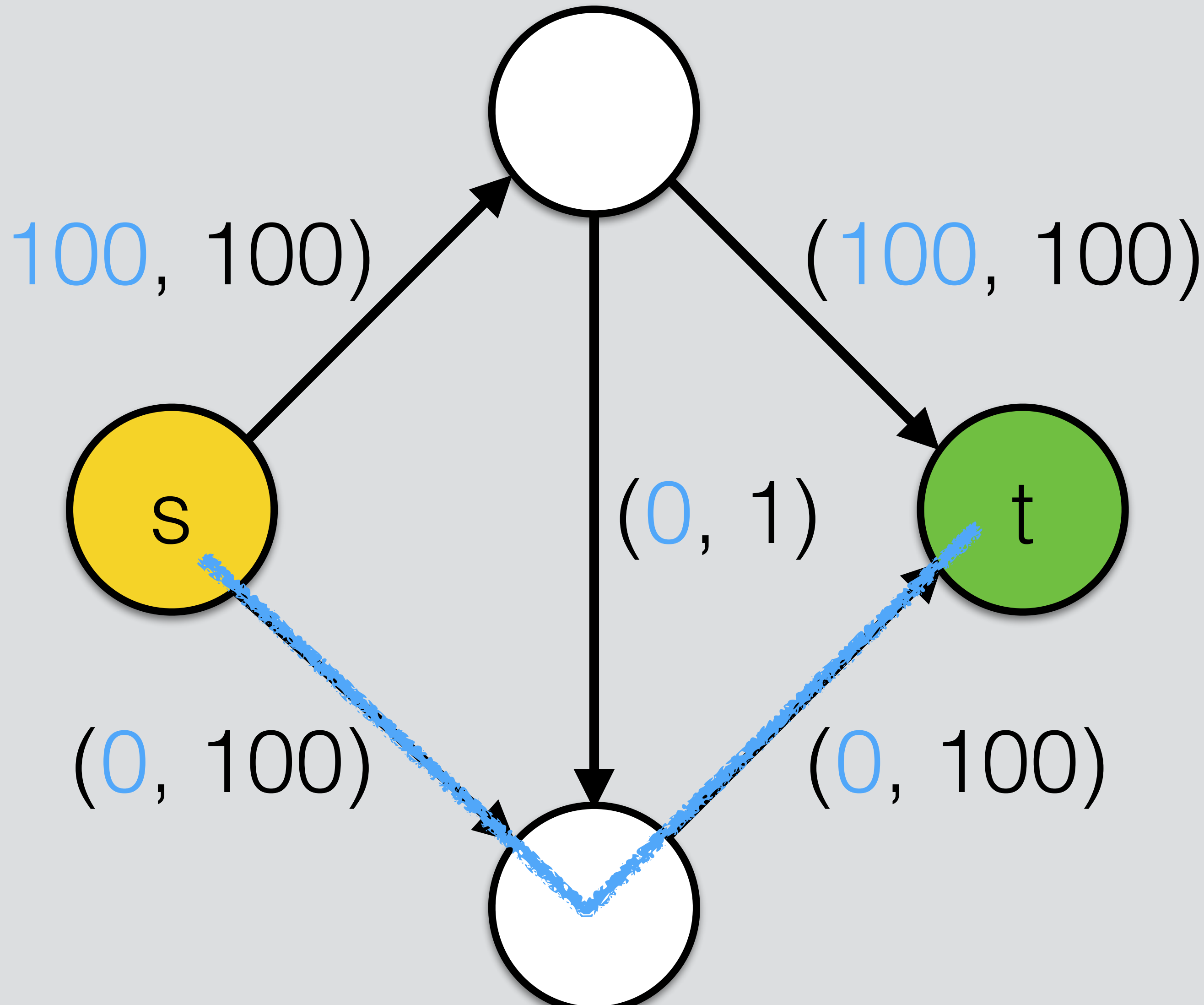
Graph (with flow)



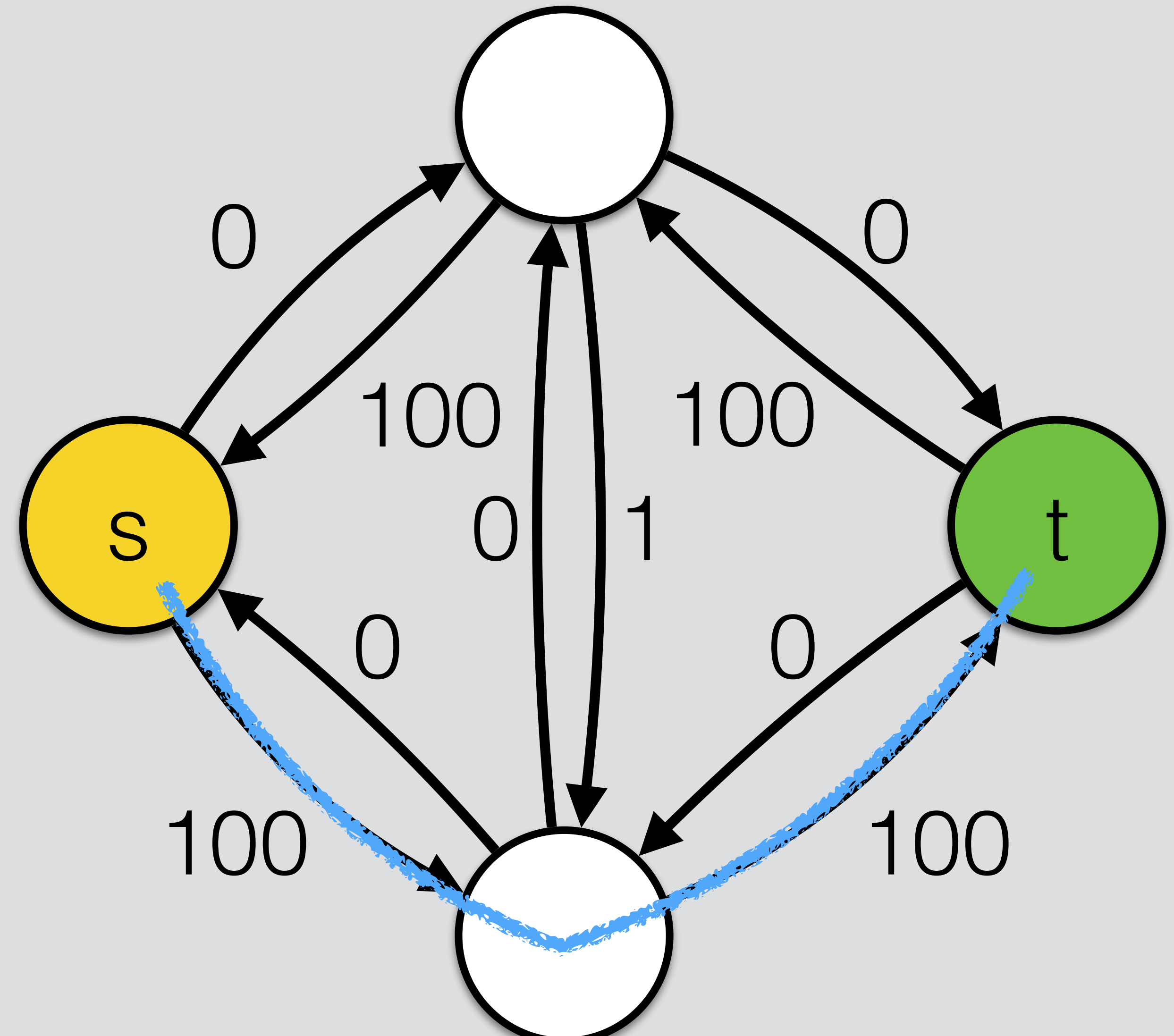
Residual graph



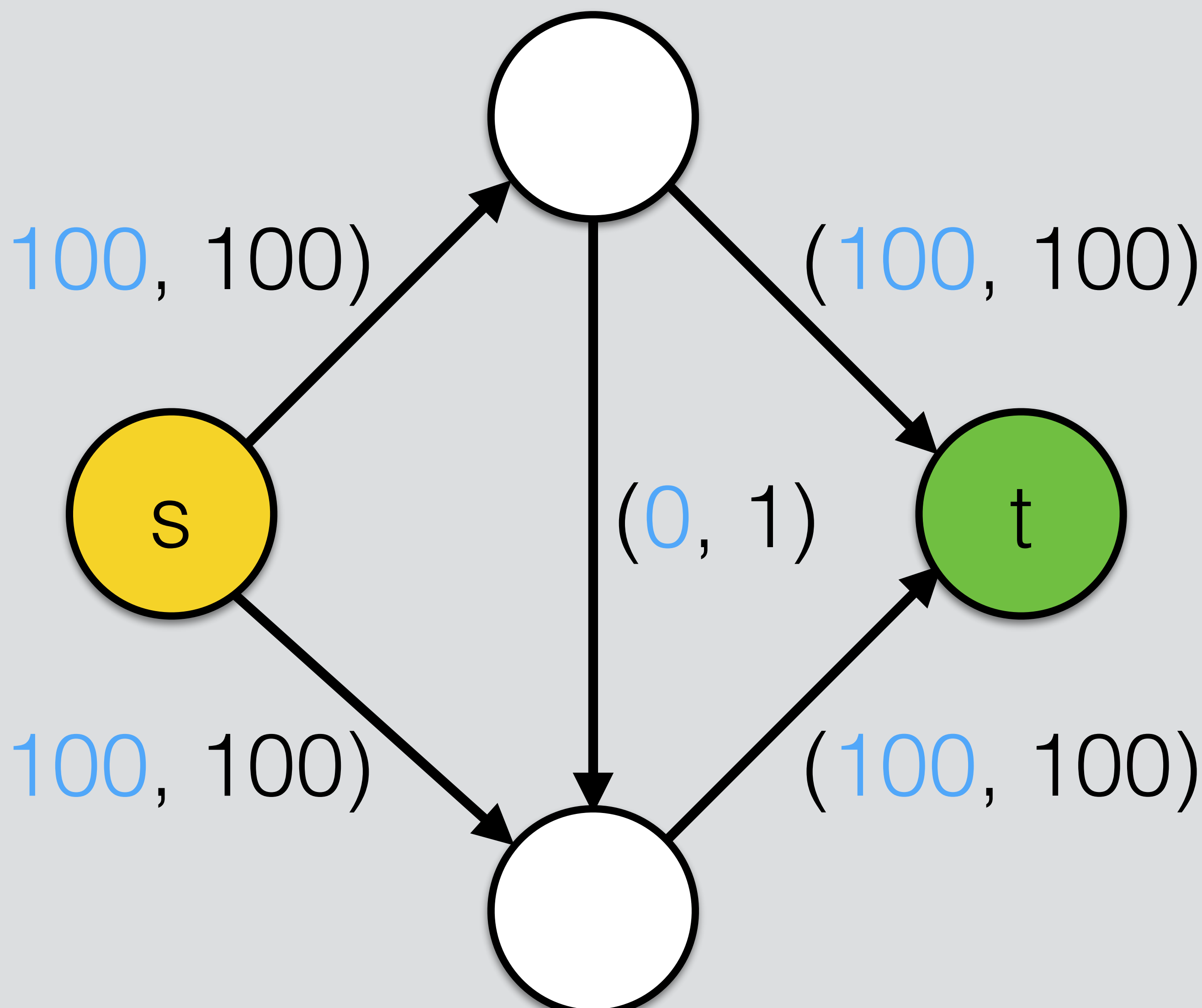
Graph (with flow)



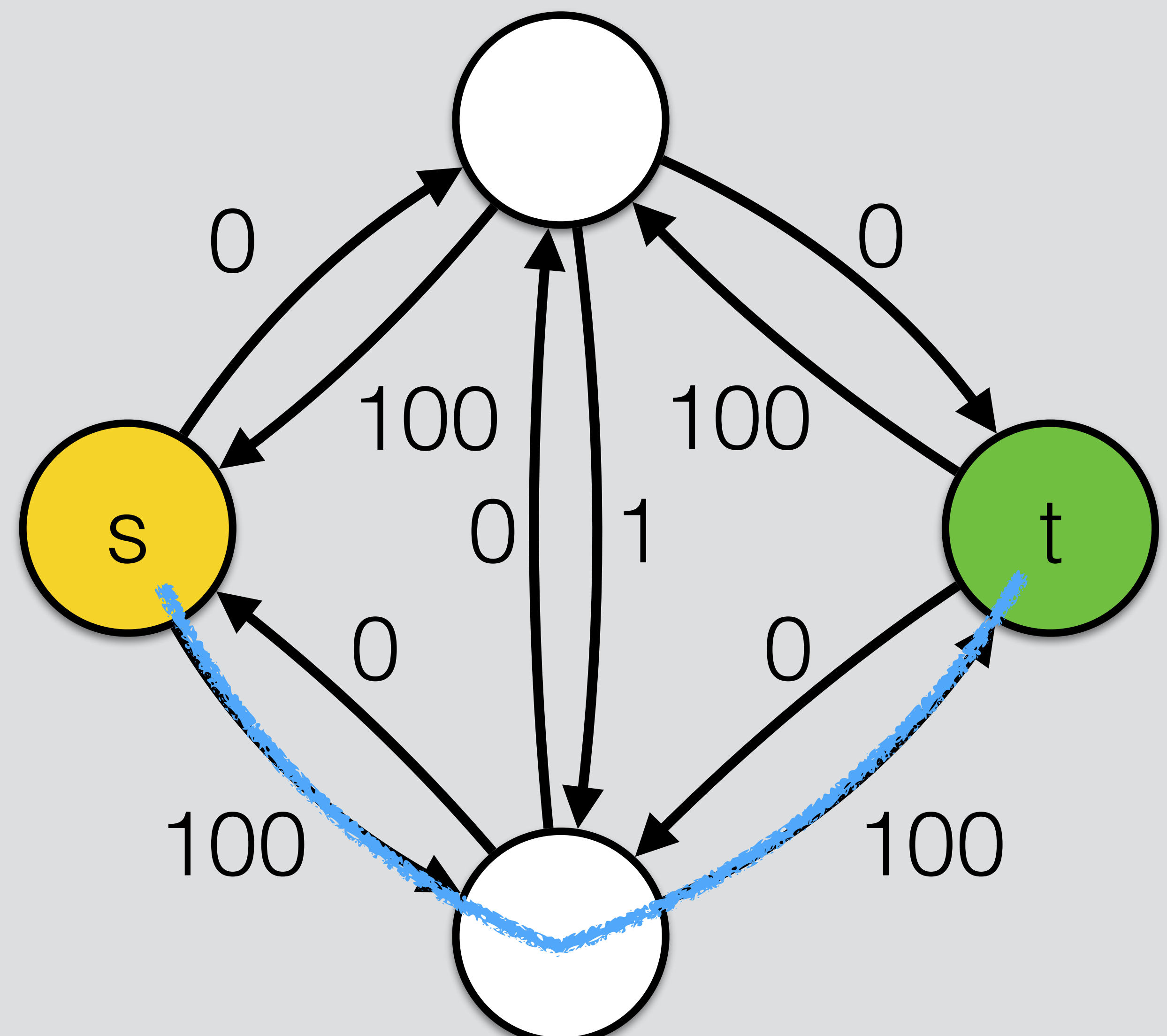
Residual graph



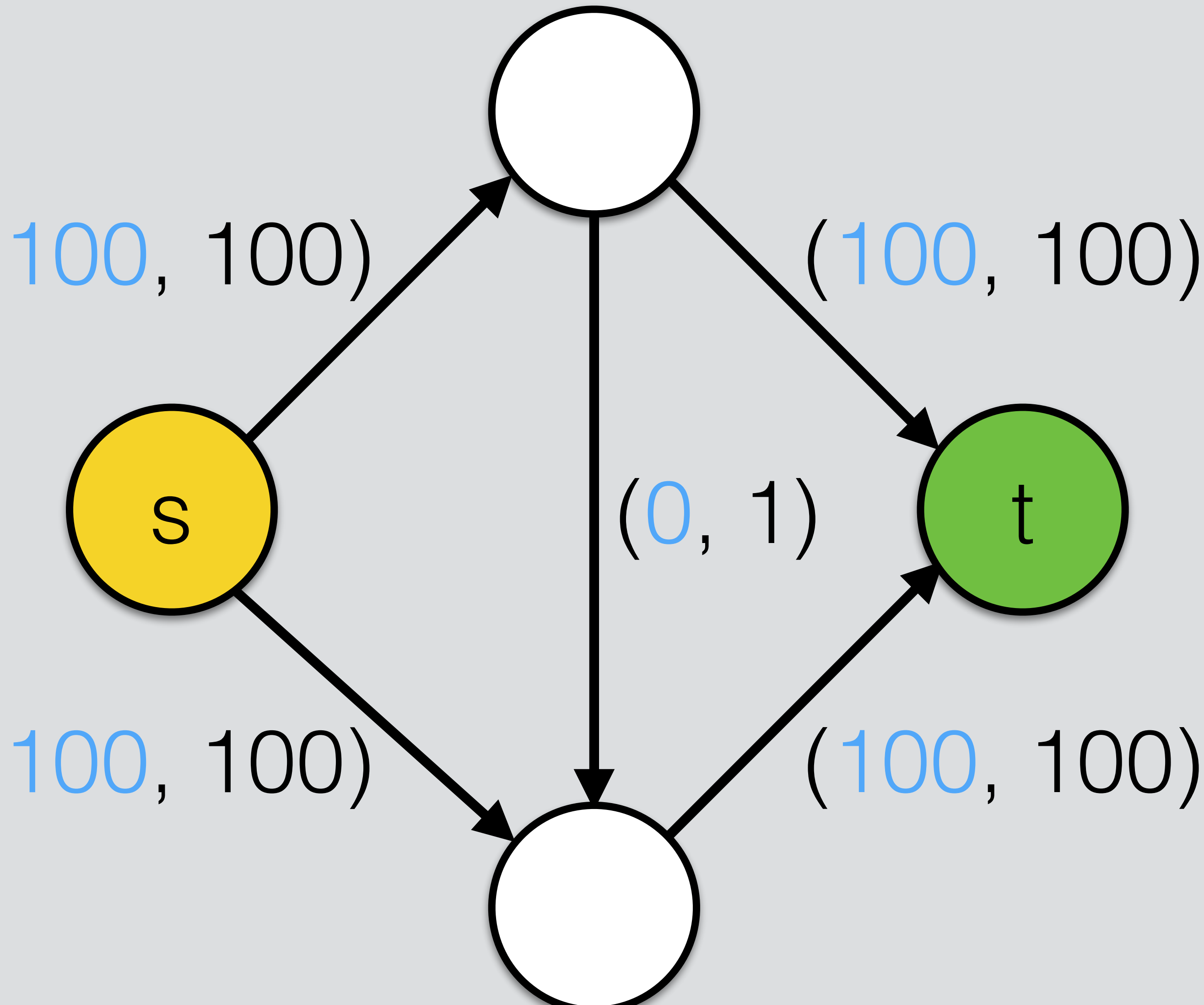
Graph (with flow)



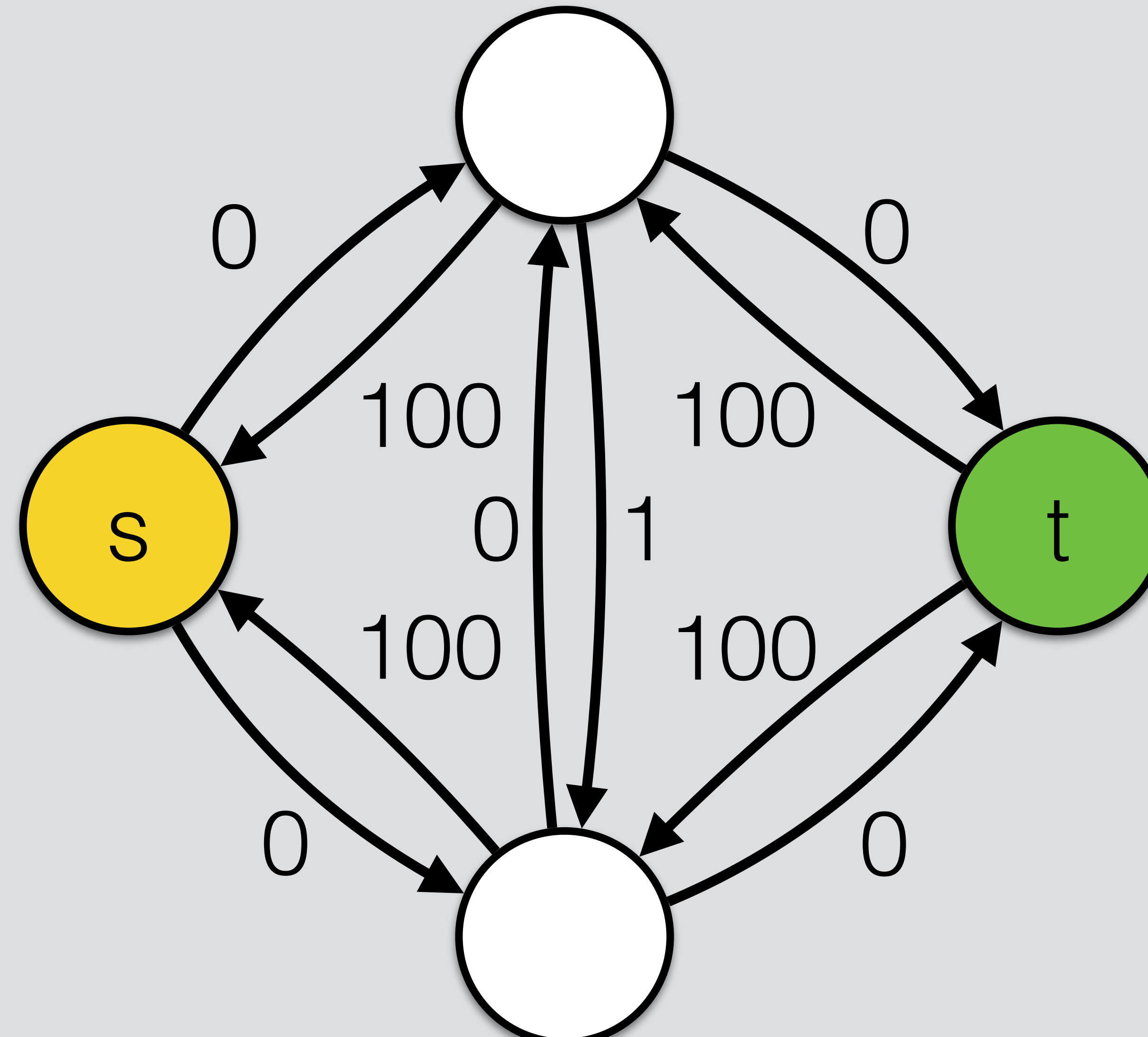
Residual graph



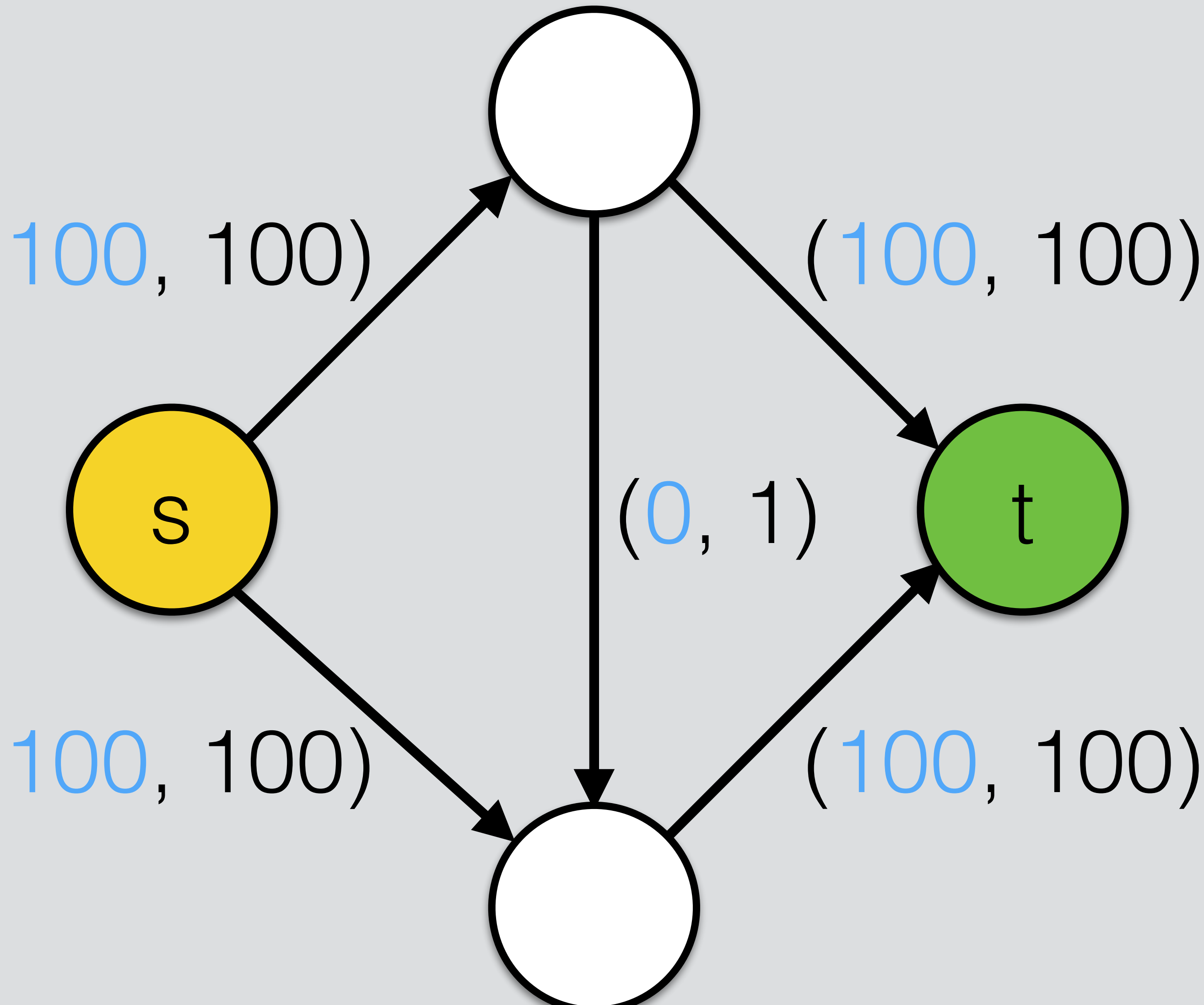
Graph (with flow)



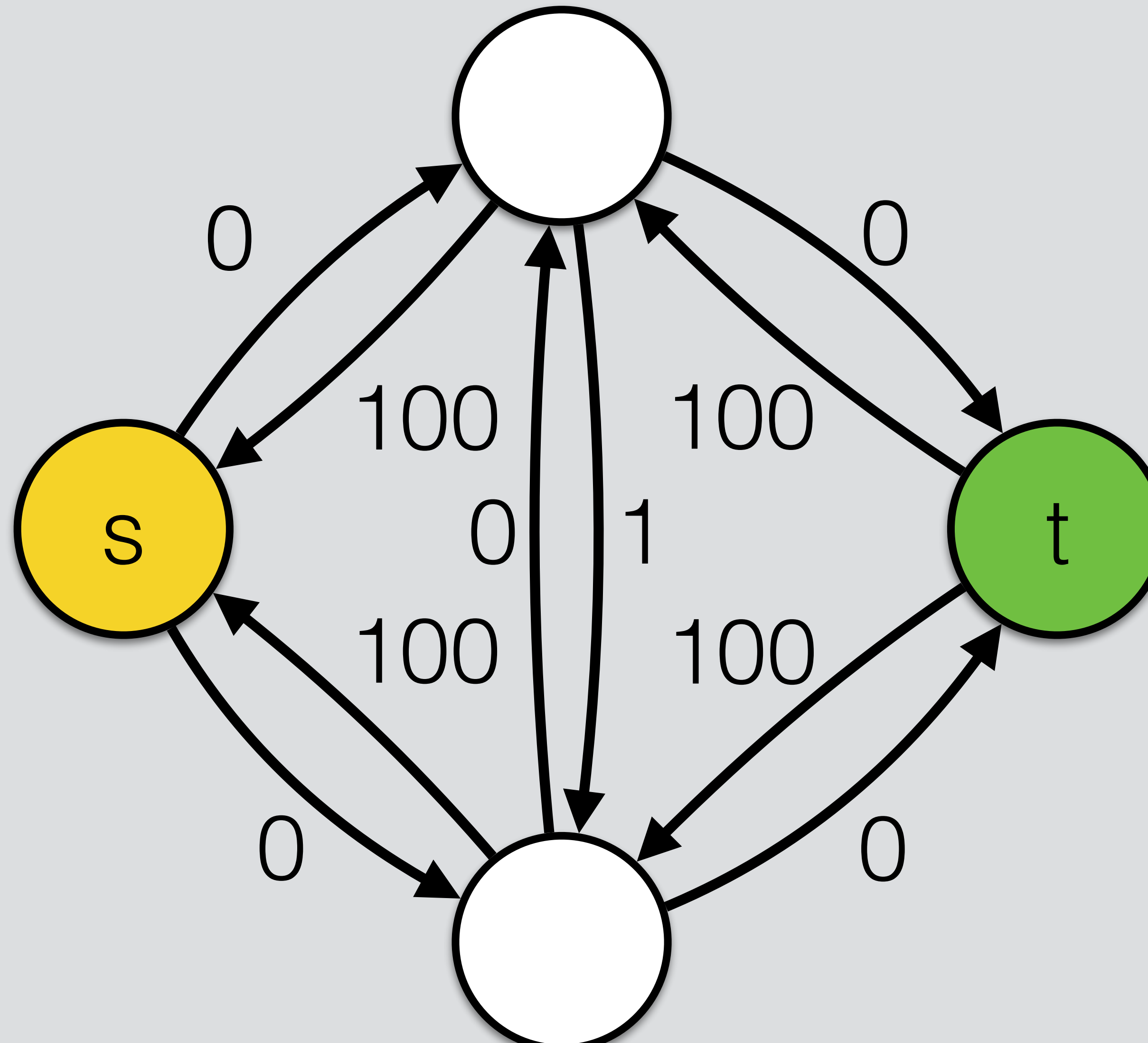
Residual graph



Graph (with flow)



Residual graph



Only 2 iterations.

Edmonds-Karp Running Time

Edmonds-Karp Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`)

Create an empty map `F`.

for (every edge (v_i, v_j) in E):

  if ((v_j, v_i) is not in `E`):

    Add (v_j, v_i) to `E` with capacity 0.

  $F[(v_i, v_j)] = 0;$

while (true):

  Let p = an (edge-length-)shortest augmenting path in residual of G.

  If no `p` exists, `break;`

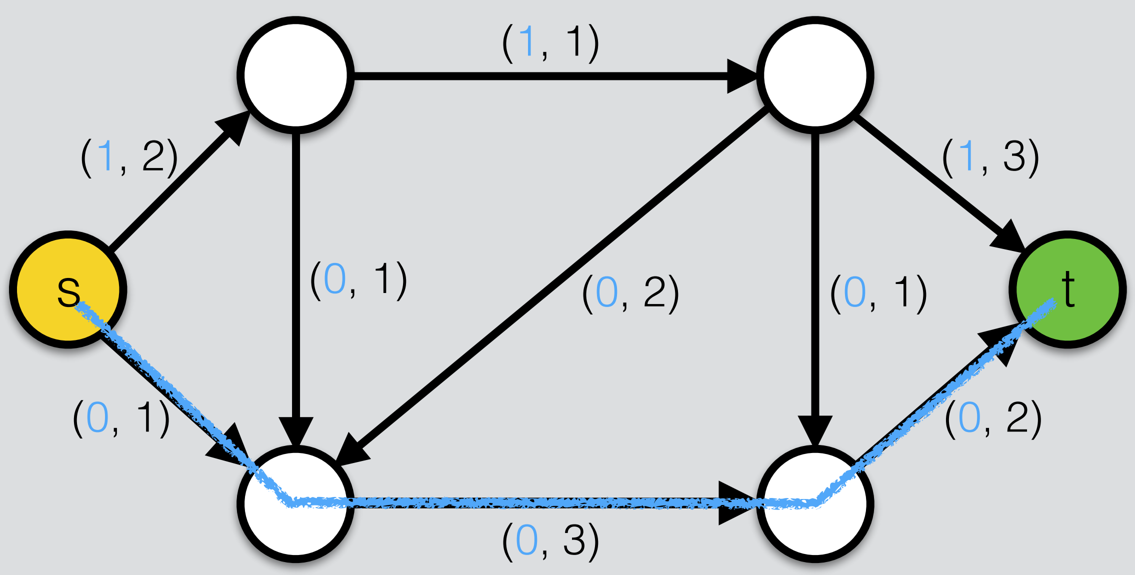
  f_m = minimum weight of edge in `p`.

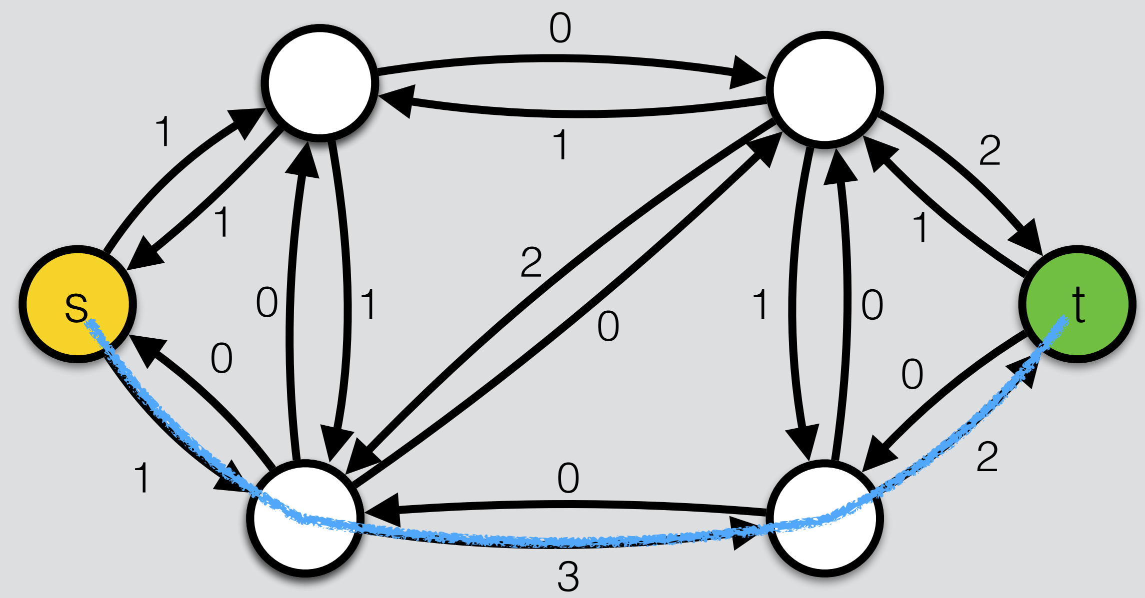
  Update flow along edges of `p` by f_m.

How many iterations?

Progress in Augmenting Paths

Lemma: each time the flow is increased, an edge reaches capacity.





Progress in Augmenting Paths

Lemma: each time the flow is increased, an edge reaches capacity.

Lemma: each time an edge (v_i, v_j) reaches capacity, the augmenting path from s to (v_i, v_j) is longer.

Progress in Augmenting Paths

Lemma: each time the flow is increased, an edge reaches capacity.

Lemma: each time an edge (v_i, v_j) reaches capacity, the augmenting path from s to (v_i, v_j) is longer.

So each edge reaches capacity $\leq$ n times.

So in total, edges reach capacity $\leq$ nm times.

So $\leq$ nm iterations.

Edmonds-Karp Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`)

Create an empty map `F`. $\Theta(n)$

for (every edge (v_i, v_j) in E):

    if $((v_j, v_i)$ is not in `E`):

        Add (v_j, v_i) to `E` with capacity 0.

    $F[(v_i, v_j)] = 0;$

$\Theta(m)$

$O(nm^2 + n^2m)$ total

while (true):

    Let p = an (edge-length-)shortest augmenting path in residual of G.

    If no `p` exists, `break;`

$\Theta(n + m)$

    f_m = minimum weight of edge in `p`.

    Update flow along edges of `p` by f_m.

$\Theta(n)$

Edmonds-Karp Algorithm

(for a graph `G = (V, E)`, source `s`, sink `t`)

Create an empty map `F`. $\Theta(n)$

for (every edge (v_i, v_j) in E):

    if $((v_j, v_i)$ is not in `E`):

        Add (v_j, v_i) to `E` with capacity 0.

    `F[(vi, vj)] = 0;` $\Theta(m)$

O((n + m)nm) time

while (true):

    Let p = an (edge-length-)shortest augmenting path in residual of G. $\Theta(n + m)$

    If no `p` exists, `break;`

    f_m = minimum weight of edge in `p`.

    Update flow along edges of `p` by f_m. $\Theta(n)$

$O(nm^2 + n^2m)$ total