Elementi di Teoria dei Grafi

Argomenti lezione:

- Il problema del massimo flusso

("*maximum flow problem*")

- Definizioni preliminari
- Condizioni di ottimalità
- Algoritmo di Ford-Fulkerson
- Esercizi

Problema del massimo flusso

- Il problema del massimo flusso consiste nel trovare il modo migliore per trasferire entità <u>da un punto a un altro in un digrafo</u>

SATURA

- Ogni arco è caratterizzato da una <u>capacità massima</u>, ovvero da un limite massimo di entità che possono attraversarlo

- Nei problemi di flusso <u>il digrafo è capacitato</u>, ovvero gli archi hanno una <u>portata massima</u> che non può essere superata, mentre nei problemi di percorso minimo gli archi hanno un peso/costo

- Nel problema di individuare il flusso di costo minimo questi due aspetti vengono entrambi presi in considerazione

- <u>Applicazioni pratiche</u>: Ad esempio nel campo dei trasporti o nel campo delle reti elettriche o altre infrastrutture critiche

Rete di flusso

Rete di flusso : è un digrafo capacitato chiamato $R = (N, A, u)$

Gli archi (i, j) in A hanno una *capacità* non negativa $u_{ij} >= 0$

A differenza di altri problemi visti (minimo albero ricoprente e percorso minimo), le etichette associate a gli archi sono delle *capacità massime* che non possono essere superate e non dei pesi o dei costi che devono essere minimizzati

In una rete di flusso definiamo *nodo sorgente s* e *nodo pozzo t*

Il problema del massimo flusso consiste nell'individuare il *massimo flusso* da s a t nella rete che rispetti tutti i vincoli dati

Vincoli di una rete di flusso (1)

Analizziamo due categorie di vincoli:

1. i vincoli sulla *capacità del flusso* che scorre negli archi
2. i vincoli sul *bilanciamento dei flussi* entranti/uscenti nei/dai nodi

In ogni soluzione ammissibile, il flusso (eventualmente nullo) x_{ij} che scorre nell' arco (i, j) non potrà mai superare il massimo flusso che può attraversare l'arco u_{ij}, ovvero non potrà mai eccedere la sua capacità (I categoria di vincoli)

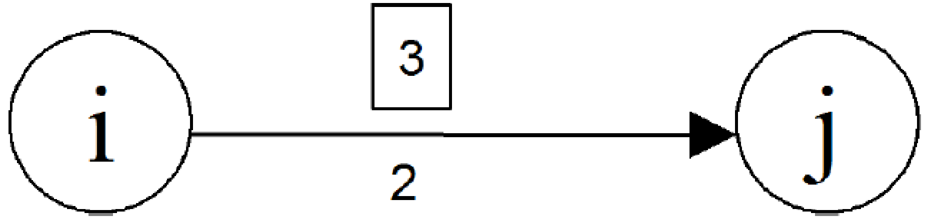
Lo scalare x_{ij} relativo all'arco (i, j) in A viene detto *valore* del flusso:

$$0 \leq x_{i,j} \leq u_{ij}, \forall (i, j) \in A$$

Esempio:

$u_{ij} = 3$

$x_{ij} = 2$



Vincoli di una rete di flusso (2)

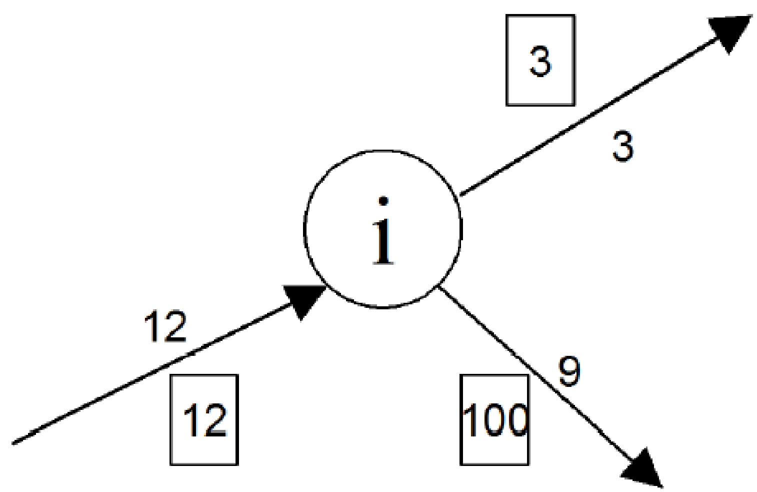
<u>II categoria di vincoli</u> sono i vincoli di *bilanciamento delle masse* ai nodi

Questi vincoli implicano che la somma dei flussi sugli archi entranti in ogni nodo (tranne *s* e *t*) deve essere <u>uguale</u> alla somma dei flussi sugli archi uscenti ovvero in ogni nodo non si può ne generare ne perdere del flusso:

$$\sum_{(j,i)\in\delta^+(i)} x_{j,i} = \sum_{(i,j)\in\delta^-(i)} x_{i,j}, \forall i \in N \setminus \{s,t\}$$

<u>Esempio</u>:
Flusso entrante in *i*: 12
=
Flusso uscente da *i*: 9+3

Capacità entrante in *i*: 12
!=
Capacità uscente da *i*: 103



Flusso ammissibile

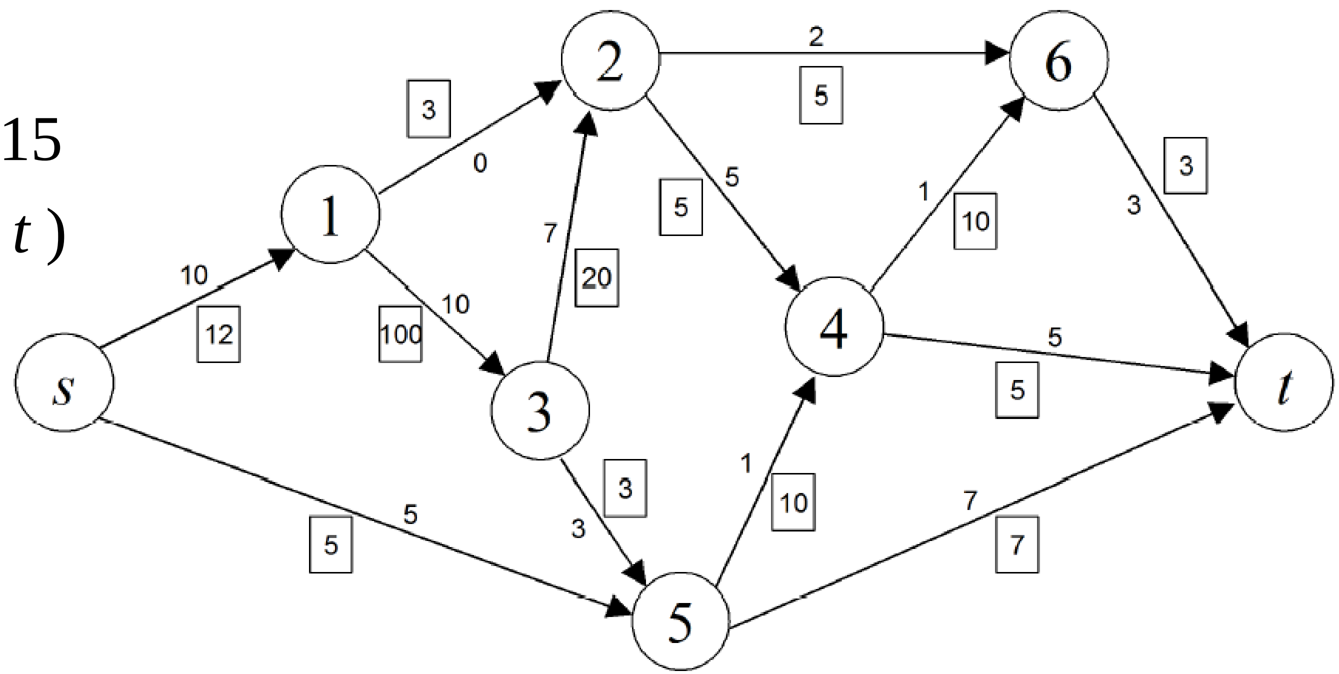
Un flusso si dice *ammissibile* se per ogni arco e per ogni nodo sono rispettate le capacità degli archi (I categoria di vincoli) e il bilanciamento delle masse ai nodi (II categoria di vincoli)

Il problema del massimo flusso consiste nell'individuare un flusso da *s* a *t* nella rete tale che <u>tutte le capacità sugli archi non siano violate e che il flusso individuato sia MASSIMO</u>

Esempio:

Flusso totale = 15

(15 unità da *s* a *t*)



Assunzioni nei problemi affrontati (1)

Rete orientata: Nel caso in cui la rete di flusso non sia orientata, è sempre possibile trasformare il grafo corrispondente in un digrafo, infatti ogni lato (non orientato) del grafo può essere sostituito con due archi (orientati diversamente) nel digrafo.

Capacità sono interi non negativi: Anche questa assunzione non è troppo limitativa infatti moltiplicando per numeri adatti è possibile riportarsi al caso in cui le capacità siano numeri interi e non negativi. Questa assunzione è necessaria per garantire la correttezza e la terminazione di alcuni algoritmi.

Assenza di cammini a capacità infinita: Nel caso in cui esista un cammino che connette il nodo sorgente al nodo pozzo composto da tutti archi con capacità infinita allora il massimo flusso che può scorrere nella rete di flusso è illimitato.

Assunzioni nei problemi affrontati (2)

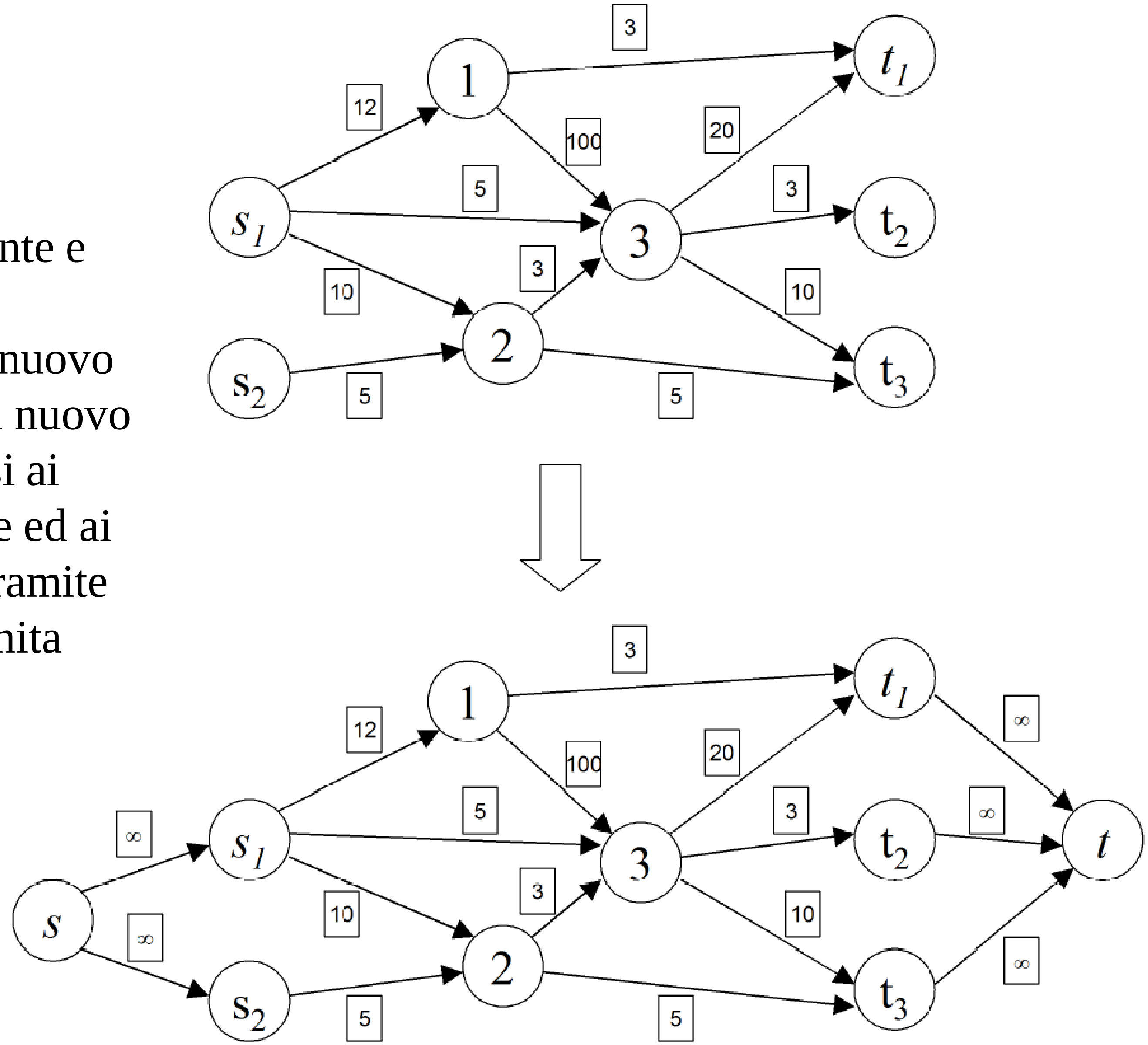
Rete bidirezionale: Se esiste un arco (i, j) allora esiste anche l'arco (j, i). Si noti come questa assunzione non sia limitativa in quanto nel caso in cui l'arco (j, i) non esista sarà comunque possibile introdurlo come un arco con capacità nulla.

Assenza di archi multipli. Questa assunzione è non restrittiva, infatti nel caso di archi multipli, che hanno stessa testa e stessa coda, allora è possibile sostituirli con un solo arco con capacità pari alla somma delle capacità di tutti gli archi multipli.

Rete con singola sorgente e singolo pozzo: Nel caso in cui esistano più nodi sorgenti o più nodi pozzo è possibile generare una nuova rete di flusso che abbia un solo nodo sorgente e un solo nodo pozzo che sia equivalente alla rete originale.

Assunzioni nei problemi affrontati (3)

Esempio:
Trasformazione di
una rete multi sorgente e
multi pozzo:
Basta introdurre un nuovo
nodo sorgente ed un nuovo
nodo pozzo connessi ai
vecchi nodi sorgente ed ai
vecchi nodi pozzo tramite
archi a capacità infinita



Capacità di un taglio

- T*aglio* $[S, \underline{S}]$ (oppure $s - t$) è una partizione di nodi negli insiemi S e $\underline{S} = N \setminus S$ tali che il nodo $s \in S$ e il nodo $t \in \underline{S}$ (in cui si ha $S \cup \underline{S} = N$ e $S \cap \underline{S} = \emptyset$)

- Gli archi che attraversano un taglio sono raggruppati in 2 set: $(S, \underline{S})$ degli archi uscenti da S e $(\underline{S}, S)$ degli archi entranti in S

- $u[S, \underline{S}]$ è la *capacità del taglio* $[S, \underline{S}]$ ovvero la somma delle capacità degli <u>archi uscenti</u> dal taglio: $$\sum_{(i,j)\in (S, \underline{S})} u_{i,j}$$

- Gli <u>archi entranti</u> nel taglio $(h, k) \in (\underline{S}, S)$ <u>non influenzano</u> la capacità del taglio $[S, \underline{S}]$

- La capacità di un taglio è un *upper bound* sul valore del massimo flusso che può effettivamente scorrere in una rete

Flussi attraverso i tagli (1)

Dato un flusso ammissibile x in una rete, si definisce *flusso netto del taglio* la quantità $v = x[S, \underline{S}]$ data dall'espressione:

$$x[S,\underline{S}] = \sum_{(i,j)\in(S,\underline{S})} x_{ij} - \sum_{(j,i)\in(\underline{S},S)} x_{ji}$$

Il primo termine dell'espressione rappresenta la quantità netta di flusso che viene inviata dai nodi S ai nodi $\underline{S}$, mentre il secondo termine memorizza il flusso che scorre nel taglio

Pertanto, $u[S, \underline{S}]$ è una limitazione superiore sul flusso netto che può essere inviato da nodi di S a nodi di $\underline{S}$.

Flussi attraverso i tagli (2)

Dato un taglio $[S, \underline{S}]$ la capacità del taglio $u[S, \underline{S}]$ è sempre maggiore o al più uguale al flusso che lo attraversa $v = x[S, \underline{S}]$

Teorema: Per ogni flusso ammissibile x e per ogni taglio $[S, \underline{S}]$ vale $v = x[S, \underline{S}] <= u[S, \underline{S}]$

Dimostrazione: Dato che $u_{ij} >= x_{ij}$; $x_{ij} >= 0$; $x_{ji} >= 0$ si ha:

$$v = \sum_{(i,j)\in(S,\underline{S})} x_{ij} - \sum_{(j,i)\in(\underline{S},S)} x_{ji} \leq \sum_{(i,j)\in(S,\underline{S})} x_{ij} \leq \sum_{(i,j)\in(S,\underline{S})} u_{ij} = u[S,\underline{S}]$$

Questo teorema implica che se $x[S, \underline{S}] = u[S, \underline{S}]$ allora il <u>flusso</u> trovato è <u>massimo</u> e il taglio trovato è a <u>capacità</u> <u>minima</u>

Inoltre, si può dimostrare che dato un massimo flusso esiste sempre un taglio minimo per cui $x[S, \underline{S}] = u[S, \underline{S}]$

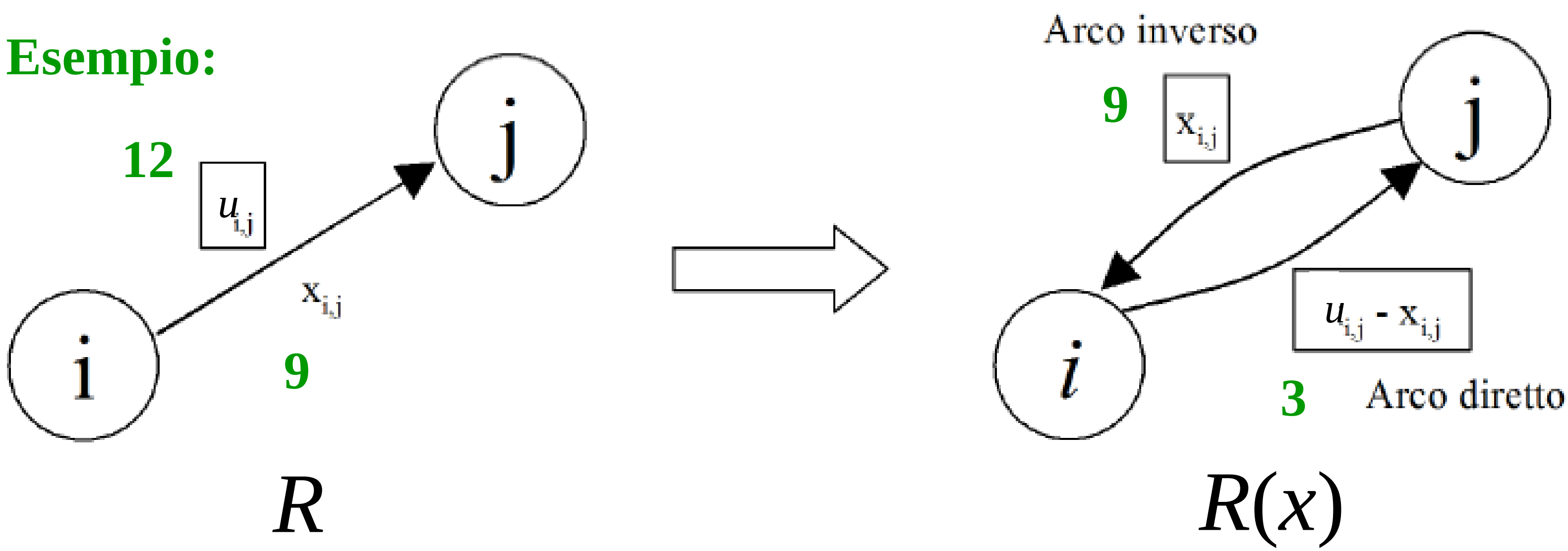
Rete residua (1)

- Dato un arco (i, j) e un flusso che lo attraversa x_{ij} definiamo:

arco saturo se $x_{ij} = u_{ij}$ mentre ***arco scarico*** se $x_{ij} = 0$

- Sia x un flusso ammissibile in una rete R, si definisce *rete residua* $R(x) = (N , \underline{A} , r)$ la rete associata alla rete originale R eliminando da $R(x)$ gli archi con capacità residua $r_{ij} = 0$
e sostituendo ogni arco (i, j) in $\underline{A}$ con due archi:
un arco *diretto* (i, j) di capacità $r_{ij} = u_{ij} - x_{ij} >= 0$
e un arco *inverso* (j, i) con capacità $r_{ji} = x_{ij} >= 0$

Esempio:

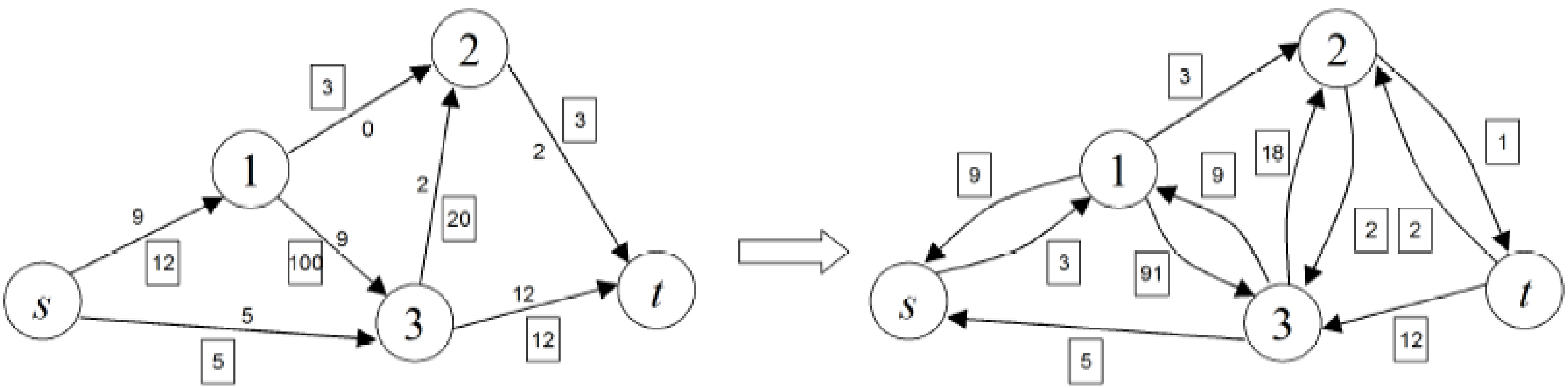


Rete residua (2)

La rete residua è definita in base al flusso ammissibile
e definisce le possibilità per variare il flusso corrente

La capacità degli archi diretti tiene conto della possibilità di incrementare il flusso corrente su un arco, mentre gli archi inversi tengono conto della possibilità di diminuire il flusso

Esempio:



R $R(x)$

Cammino aumentante (1)

Data una rete residua $R(x)$, si definisce *cammino aumentante* un cammino che congiunge il nodo sorgente s e il nodo pozzo t attraversando sia archi diretti sia archi inversi

Dato un cammino aumentante P, che congiunge il nodo s al nodo t, *è* possibile far scorrere attraverso esso un flusso δ

Il massimo flusso δ che può scorrere nel cammino aumentante sarà dato dalla più piccola capacità tra gli archi attraversati, ovvero $\boldsymbol{\delta = \min\{ r_{ij} : (i, j) \text{ in } P \} > 0}$, avendo indicato con P l'insieme degli archi appartenenti al cammino orientato da s a t

Cammino aumentante (2)

Studiamo gli effetti di un flusso δ che scorre in un cammino aumentante e le sue conseguenze nella rete (digrafo) originale *R*.

In generale:

- Attraversare un arco diretto con un flusso δ comporta nella rete originale *R* l'incremento del flusso su tale arco ($x_{ij} = x_{ij} + \delta$)

- Attraversare un arco inverso con un flusso δ comporta nella rete originale *R* la diminuzione del flusso su tale arco ($x_{ij} = x_{ij} - \delta$)

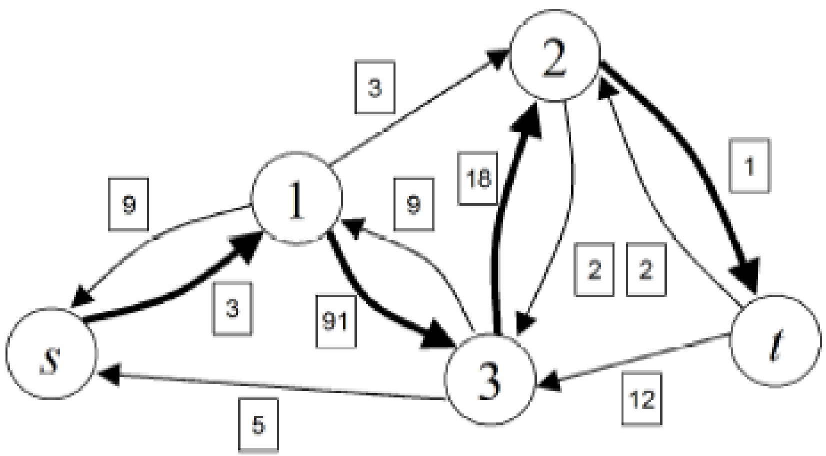
Casi limite:

Un arco saturo in *R* **può solo diminuire** (arco inverso in $R(x)$)

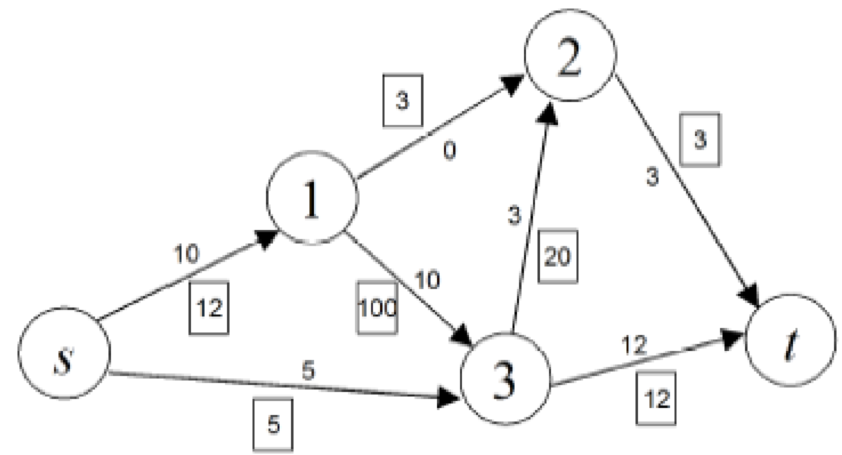
Un arco scarico in *R* **può solo aumentare** (arco diretto in $R(x)$)

Cammino aumentante (3)

Esempio: Flusso di valore 1 (si passa da flusso 14 a flusso 15)



$R(x)$

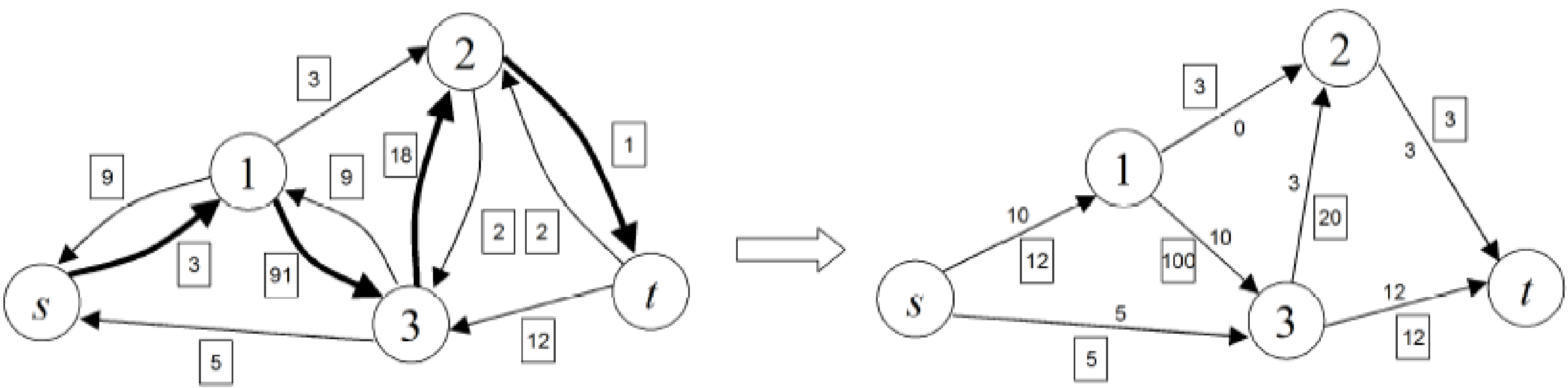


R

- Si noti come il cammino aumentante sia composto esclusivamente da archi diretti, per cui il flusso sugli archi aumenta sempre di $\delta = 1$
- Se nel cammino aumentante ci fossero stati archi inversi, allora il flusso sarebbe diminuito di δ unità su quella tipologia di archi

Condizioni di ottimalità (1)

Teorema: Una distribuzione di flusso x è <u>ottima</u> in una rete R se e solo se la rete residua $R(x)$ <u>non contiene alcun cammino orientato</u> dal nodo sorgente s al nodo pozzo t.



$R(x)$ R

Condizioni di ottimalità (2)

Dimostrazione (condizione necessaria):

Procediamo per assurdo supponendo che in corrispondenza della distribuzione di flusso ottima x con valore di flusso v, sia possibile individuare un cammino orientato P dal nodo s al nodo t nella rete residua $R(x)$.

Sia $\delta = \min\{ \underline{r}_{ij} : (i, j) \text{ in } P \} > 0$ il massimo valore del flusso aumentante. Allora è possibile ottenere un flusso in R di valore $v' = v + \delta > v$ semplicemente aumentando il flusso sugli archi diretti $x_{ij} + \delta$ e diminuendo il flusso sugli archi inversi $x_{ij} - \delta$. Pertanto, x non è una soluzione ottima, cosa che è in evidente contraddizione con l'ipotesi.

Condizioni di ottimalità (3)

Dimostrazione (condizione sufficiente):

Supponiamo che **non esista** un cammino aumentante in $R(x)$, allora deve esistere un taglio $[S, \underline{S}]$ nella rete residua con flusso nullo, $v = x[S, \underline{S}] = 0$. Si vuole dimostrare che v è ottima.

Nella rete originale R il taglio $[S, \underline{S}]$ sarà composto <u>solo da archi saturi</u> : gli archi (i, j) uscenti dal taglio aventi $x_{ij} = u_{ij}$ <u>e archi scarichi</u> : gli archi (j, i) entranti nel taglio aventi $x_{ji} = 0$. Ne consegue che per tale taglio vale:

$$v = \sum_{(i,j)\in(S,\underline{S})} x_{ij} - \sum_{(j,i)\in(\underline{S},S)} x_{ji}$$

$$v = \sum_{(i,j)\in(S,\underline{S})} u_{ij} \qquad = u[S,\underline{S}]$$

Dato che $v \le u[S, \underline{S}]$ si ha che tale flusso è ottimo.

Condizioni di ottimalità (4)

Teorema *Max-Flow Min-Cut*:

In una rete di flusso capacitata il massimo valore di flusso netto inviabile dal nodo sorgente *s* al nodo pozzo *t* è **uguale** alla capacità del minimo taglio *s*–*t*

$$v^* = \sum_{(i,j)\in(S,\underline{S})} x_{ij}^* - \sum_{(j,i)\in(\underline{S},S)} x_{ji}^*$$

$$v^* = \sum_{(i,j)\in(S,\underline{S})} u_{ij} \qquad = u[S^*,\underline{S}^*]$$

Condizioni di ottimalità (5)

Condizione di dualità: Il problema di individuare il "massimo flusso" può essere considerato alla stregua del *problema primale*, mentre il problema di individuare il "taglio di capacità minima" come il suo *problema duale*.

Dualità debole: imporre che la soluzione del primale sia sempre non superiore alla soluzione del problema duale

Dualità forte: imporre l'uguaglianza tra i valori delle soluzioni ottime dei due problemi (teorema Max-Flow Min-Cut)

Il seguente algoritmo consente quindi di risolvere contestualmente entrambi i problemi (primale e duale).

Algoritmo di Ford–Fulkerson (1)

Algoritmo di Ford–Fulkerson (Ver. $O(nmK)$**)**

// Inizializzazione

$v = 0;$

Costruisci $R(x)$ dalla distribuzione di flusso x ;

// Ciclo principale

while ($\exists P$ cammino aumentante da s a t)

{

$\delta = \min\{ r_{i,j} : (i,j) \in P\};$ (massimo flusso aumentante)

Aggiorna il flusso in R, e ricalcola $R(x)$;

$v += \delta;$ (massimo flusso corrente)

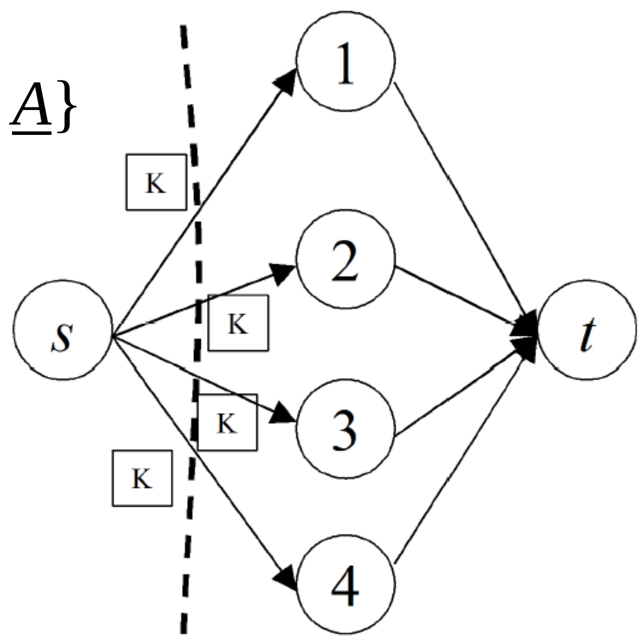
}

Algoritmo di Ford–Fulkerson (2)

<u>Complessità computazionale dell'algoritmo</u> $O(nmK)$:
O(m) : per l'operazione che viene effettuata durante ogni iterazione che è la ricerca di un cammino tra s e t nella rete $R(x)$

O(nK) : per il numero di iterazioni necessarie, nel caso peggiore, per raggiungere il massimo flusso,
con n = numero di nodi e
K = arco a capacità massima = max{u_{ij} in $\underline{A}$}

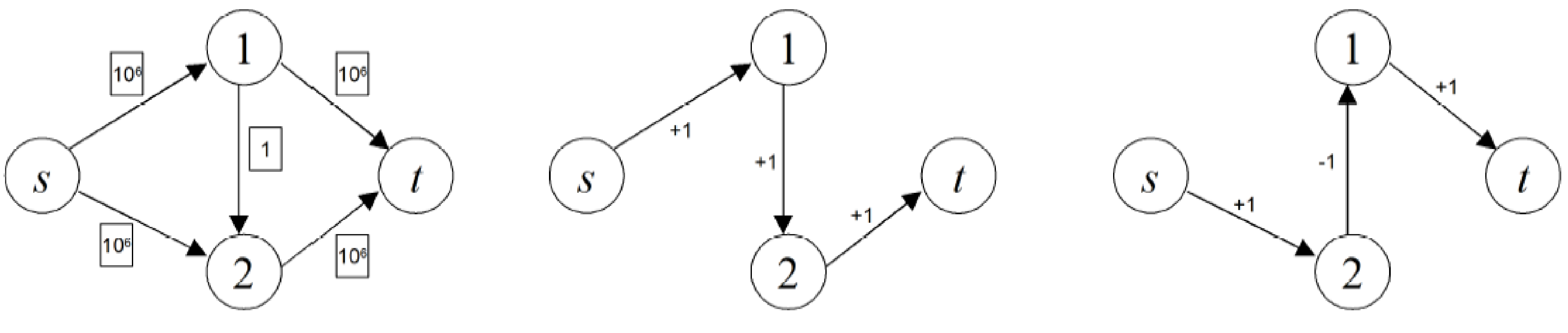
Questo algoritmo è, quindi,
non polinomiale nella dimensione dell'istanza, vista la dipendenza da K



Algoritmo di Ford–Fulkerson (3)

Istanze patologiche: numero arbitrariamente elevato di iterazioni per questa versione (basilare) dell'algoritmo

Esempio (fig. a):
La selezione alternata dei cammini $s-1-2-t$ e $s-2-1-t$ ad ogni iterazione corrisponde l'aumento del flusso di una singola unità (fig. b,c). Ma considerando che il massimo flusso è M, numero elevato, saranno necessarie $2M$ iterazioni prima di saturare la rete.
Si noti invece, come la scelta dei cammini aumentanti $s-1-t$ prima e $s-2-t$ poi, permetterebbe di saturare la rete residua in due iterazioni.



(a) (b) (c)

Algoritmo di Ford–Fulkerson (4)

<u>Svantaggio principale dell'algoritmo</u>: doversi ricalcolare ad ogni passo la rete residua $R(x)$.

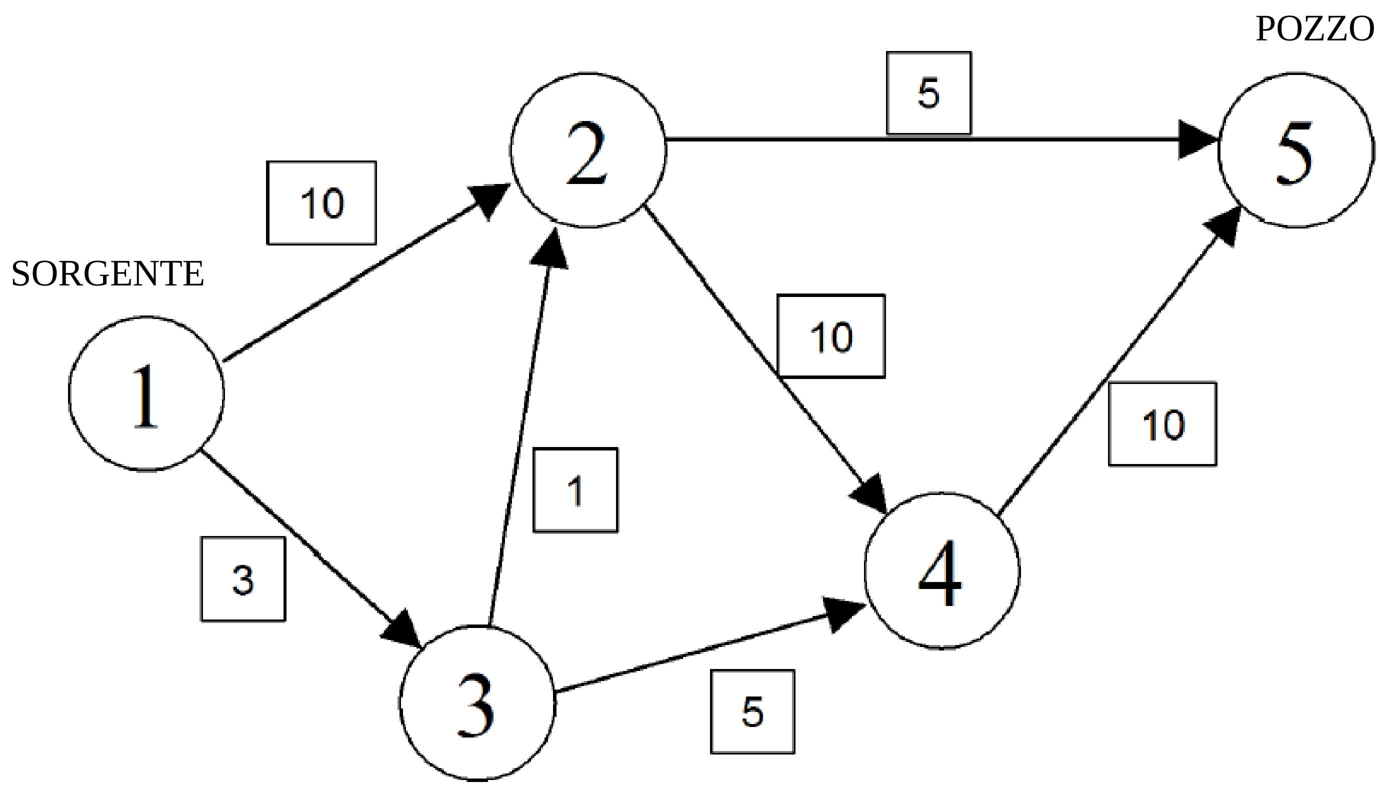
Come fare per non ricalcolare ad ogni iterazione $R(x)$?

Si possono usare due etichette [[j], ± *pred*[j]] associate ad ogni nodo al fine di individuare <u>un albero dei cammini aumentanti</u>:

- [j] memorizza il flusso che può essere portato in j sul cammino aumentante da s a j con massimo flusso
- *pred*[j] indica il nodo predecessore del nodo j sul cammino aumentante da s a j con massimo flusso
- ± *pred*[j] indica il verso del flusso tra *pred*[j] e j (+ indica un flusso diretto, − indica un flusso inverso)

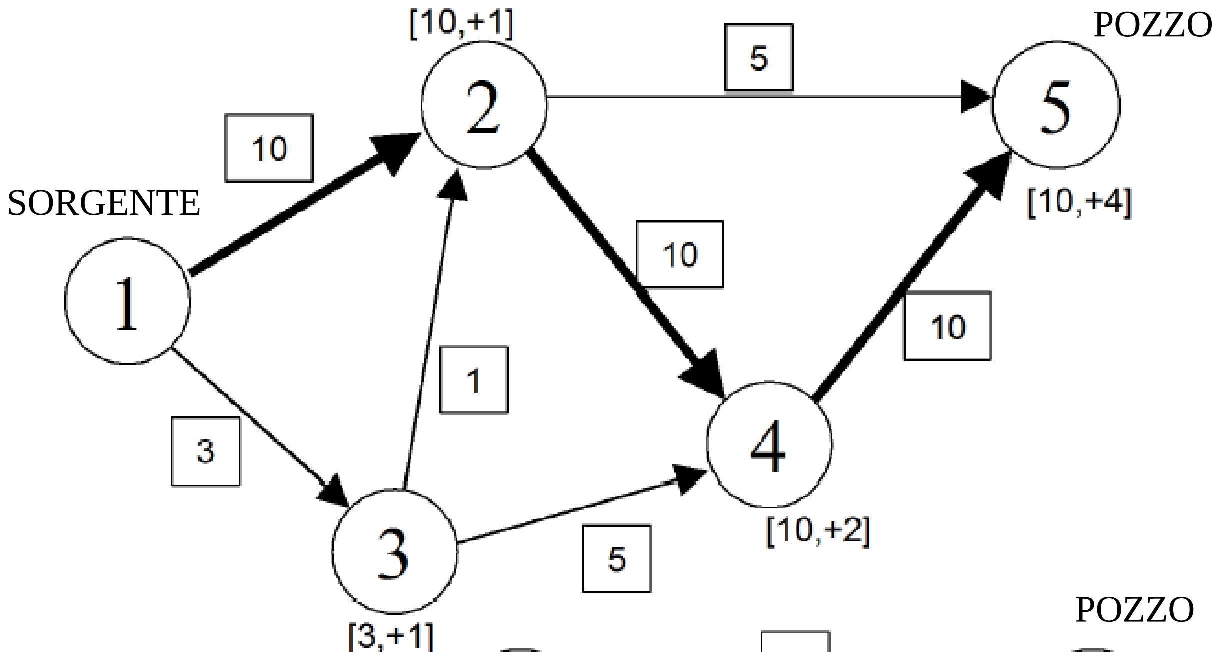
Algoritmo di Ford–Fulkerson (5)

<u>Esempio</u>: Si consideri la rete di flusso rappresentata in figura con valore di flusso iniziale nullo (v = 0)

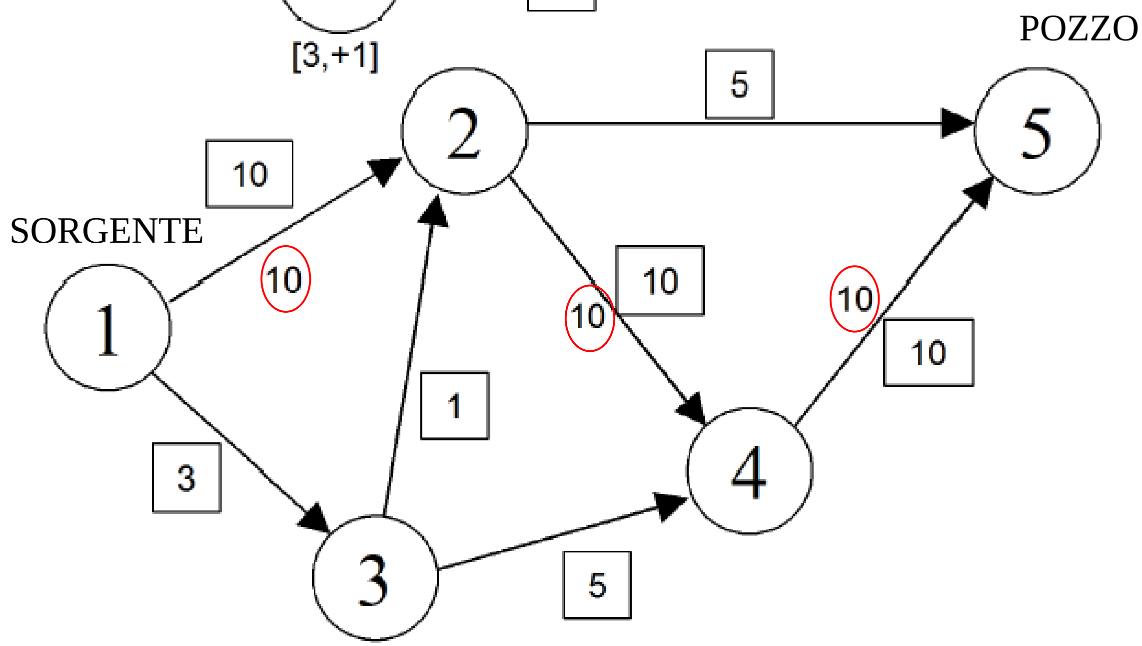


Algoritmo di Ford–Fulkerson (6)

Nella I iterazione abbiamo il cammino aumentante 1−2−4−5 con $\delta = 10$ che aumenta il flusso nella rete (v = 10)

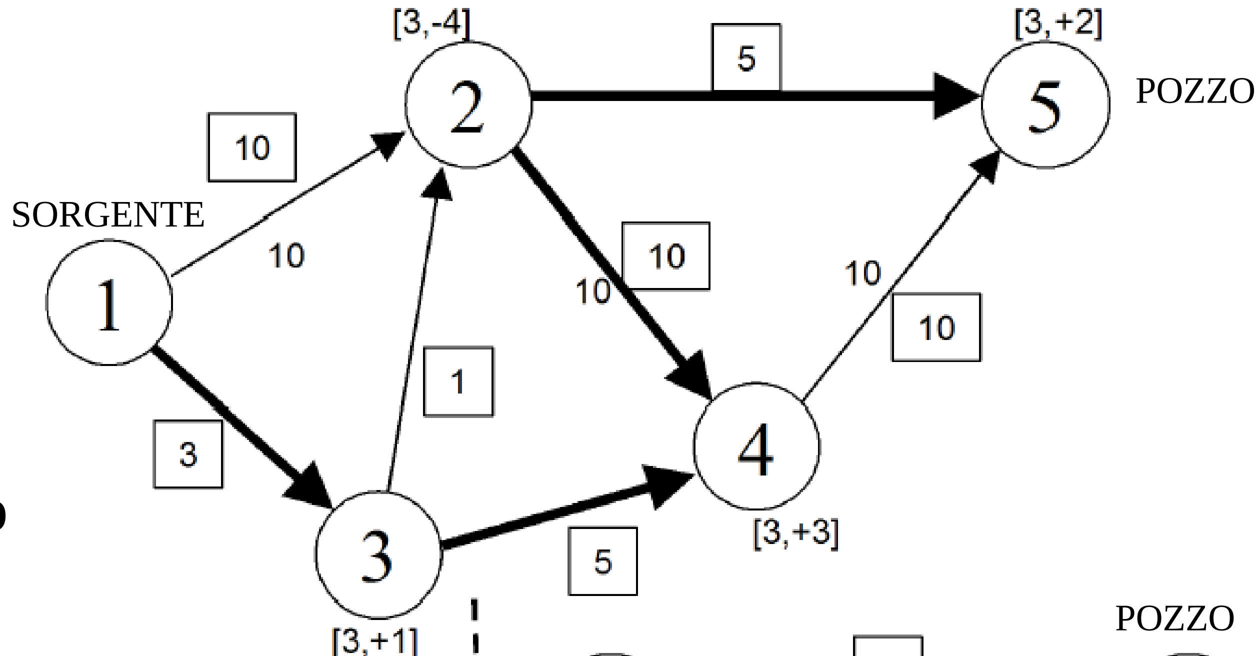


Per il nodo j = 2 si ha:
$[j]$ = 10
(flusso pari a 10 unità)
$pred[j]$ = +1
(nodo 1 e flusso diretto)



Algoritmo di Ford–Fulkerson (7)

Nella II iterazione abbiamo il cammino aumentante 1−3−4−2−5 con $\delta = 3$ che aumenta il flusso nella rete (v = 13)

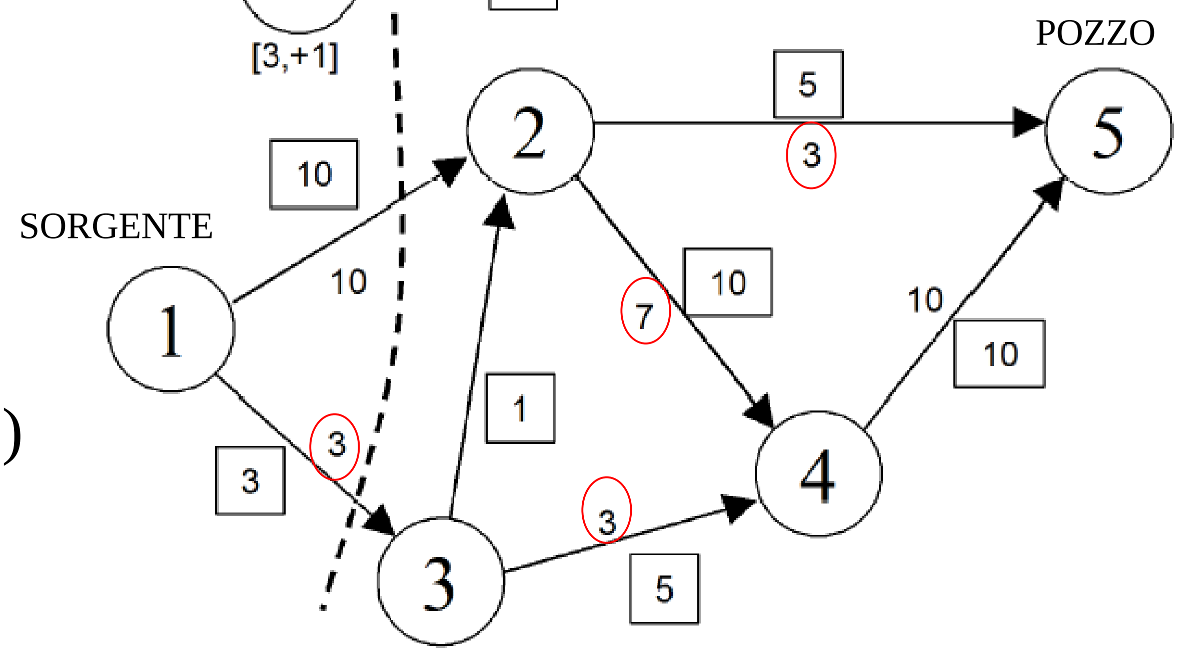


Per il nodo j = 2 si ha:

$[j] = 3$

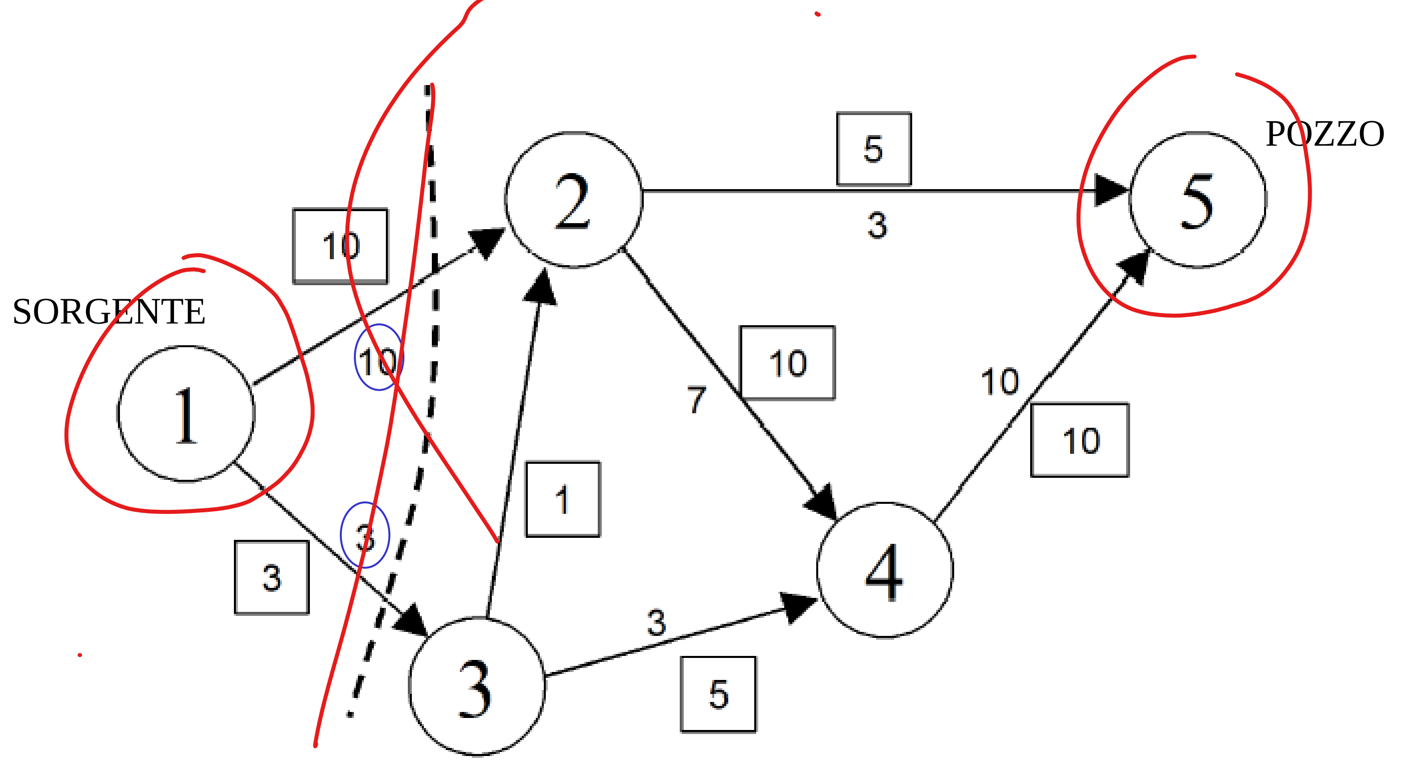
$pred[j] = -4$

(nodo 4 e flusso inverso)



Algoritmo di Ford–Fulkerson (8)

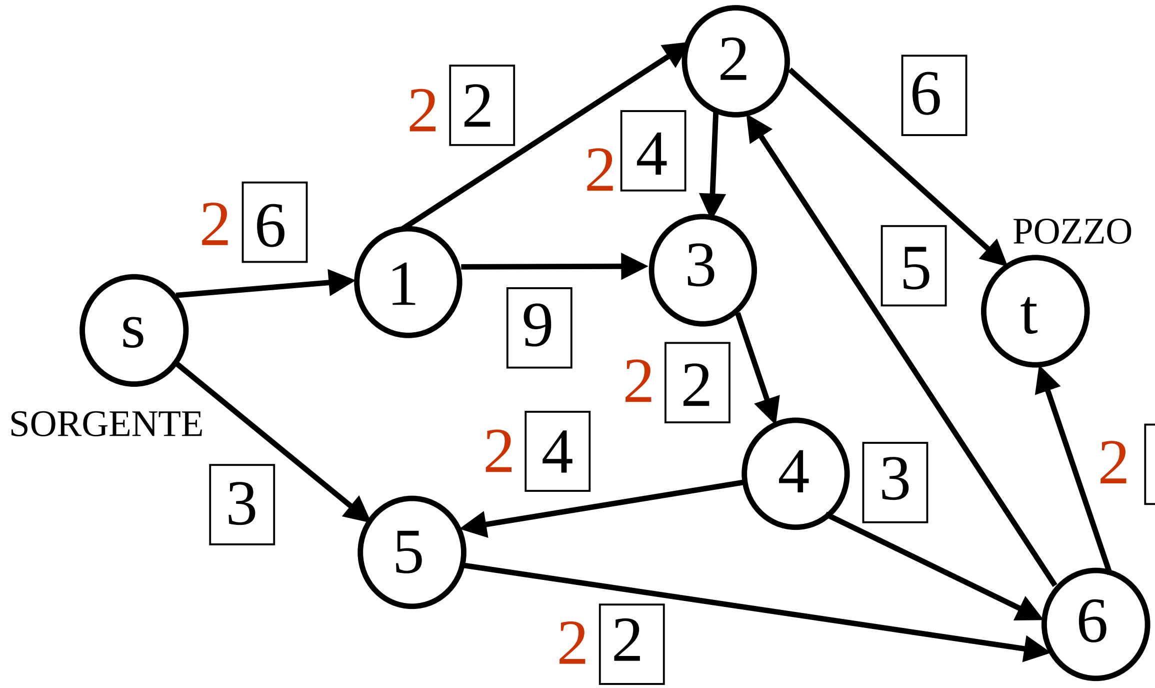
In figura viene evidenziato il taglio di capacità minima, che ha tutti gli archi da esso uscenti saturi, da cui possiamo certificare l'ottimalità della soluzione (v = 13 = ottimo)



Esercizi Algo di Ford–Fulkerson (1)

Es. 1: A partire dalla distribuzione di flusso data trovare il massimo flusso inviabile da ***s*** a ***t*** con l'algoritmo di Ford e Fulkerson:

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2



Esercizi Algo di Ford–Fulkerson (2)

Es. 1: A partire dalla distribuzione di flusso data trovare il massimo flusso inviabile da ***s*** a ***t*** con l'algoritmo di Ford e Fulkerson:

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2

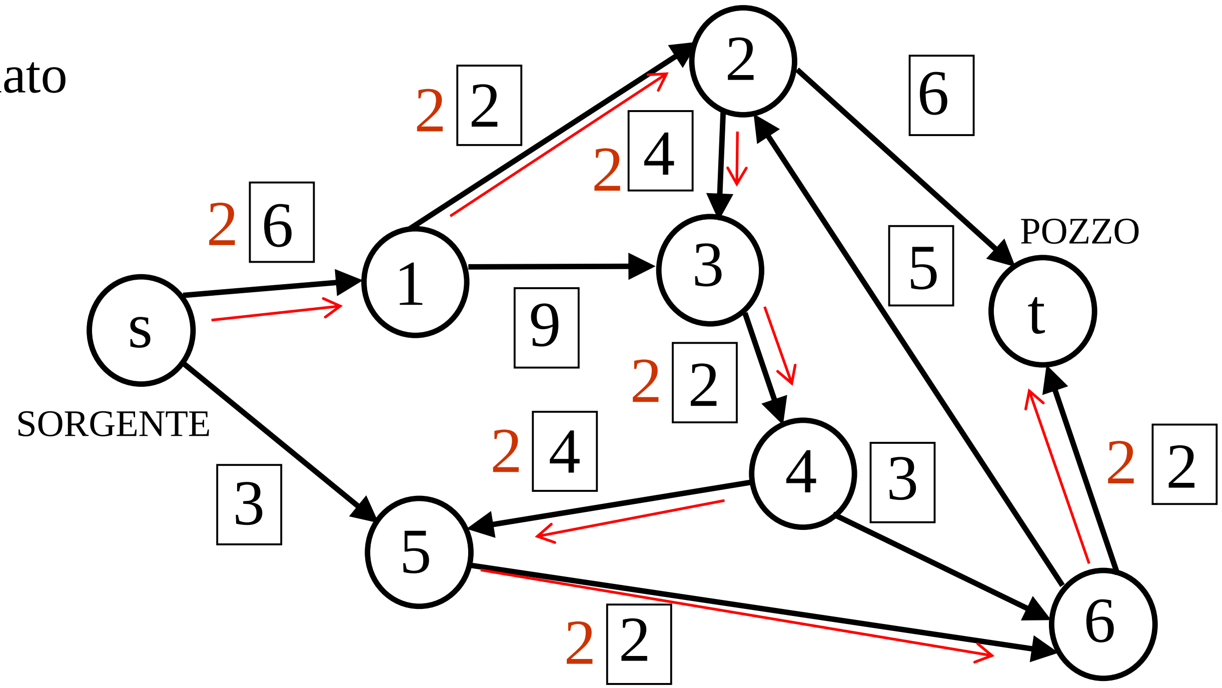
Il flusso iniziale inviato da *s* a *t* è pari a 2

massimo flusso corrente:

$\delta = 2$

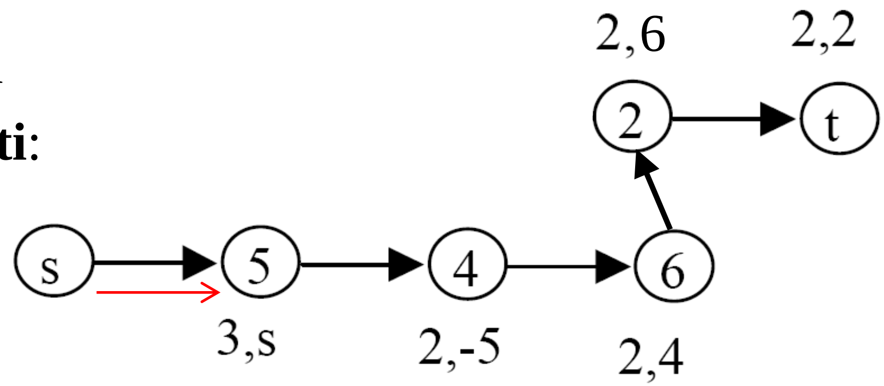
flusso complessivo:

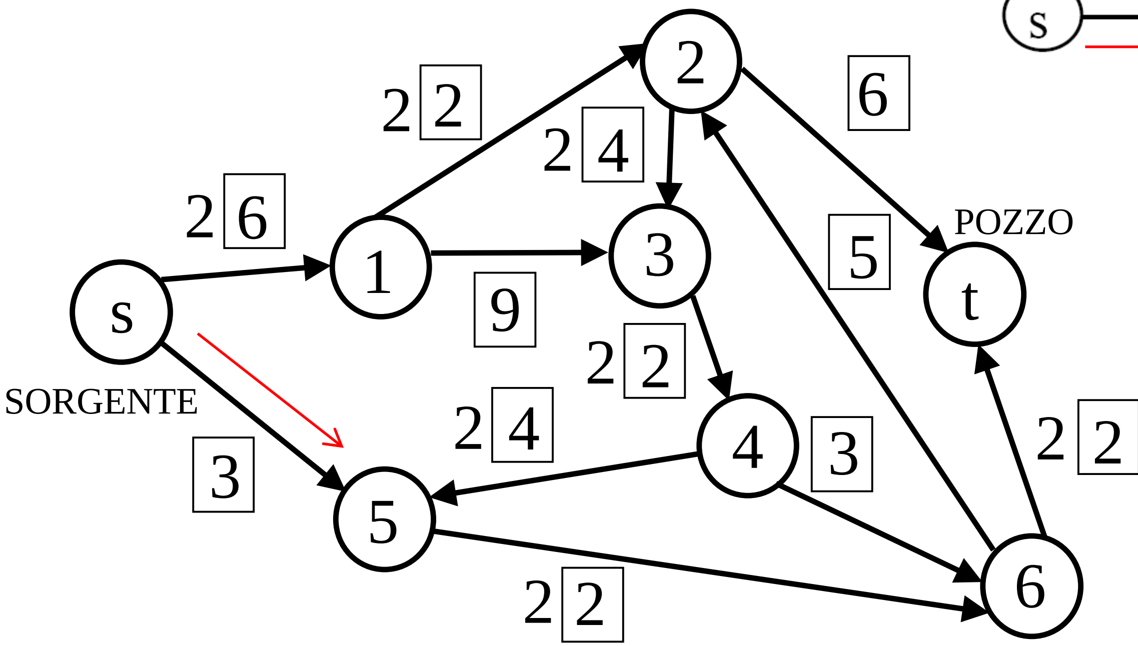
$v = 2$



Esercizi Algo di Ford–Fulkerson (3)

La ricerca del I cammino aumentante porta a costruire un **albero dei cammini aumentanti**:

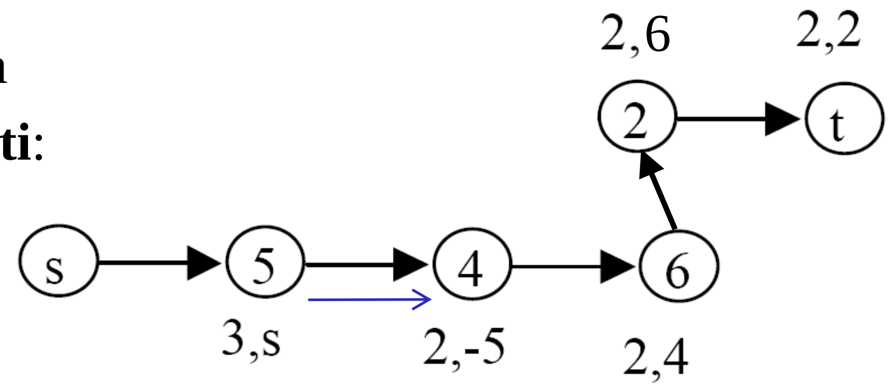


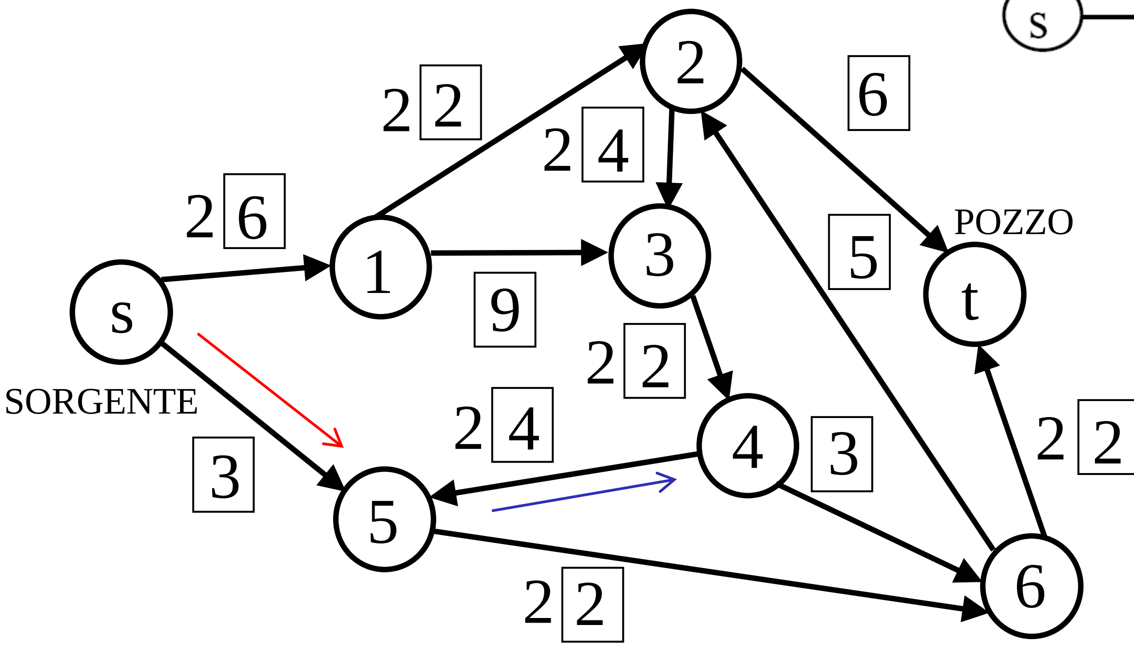


Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2

Esercizi Algo di Ford–Fulkerson (4)

La ricerca del I cammino aumentante porta a costruire un **albero dei cammini aumentanti**:

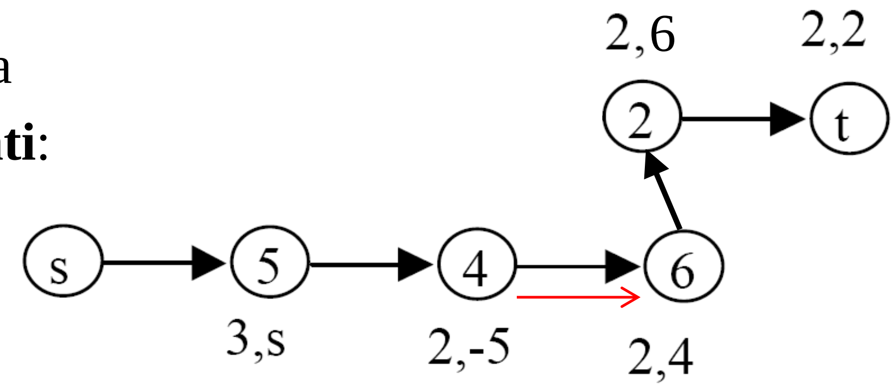


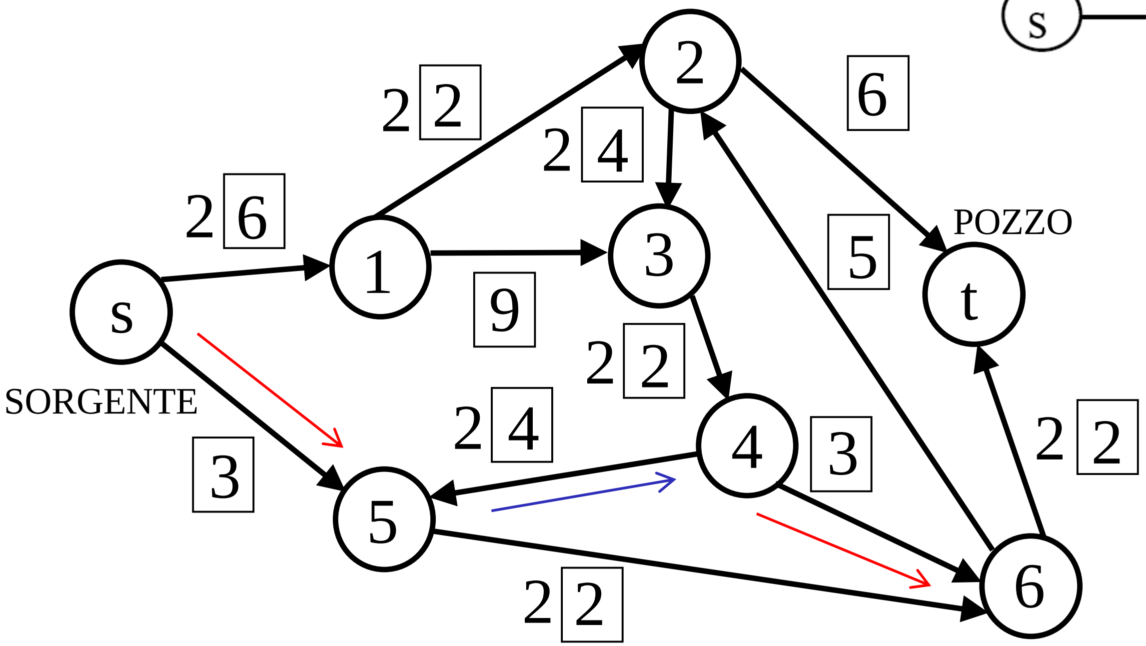


Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2

Esercizi Algo di Ford–Fulkerson (5)

La ricerca del I cammino aumentante porta a costruire un **albero dei cammini aumentanti**:

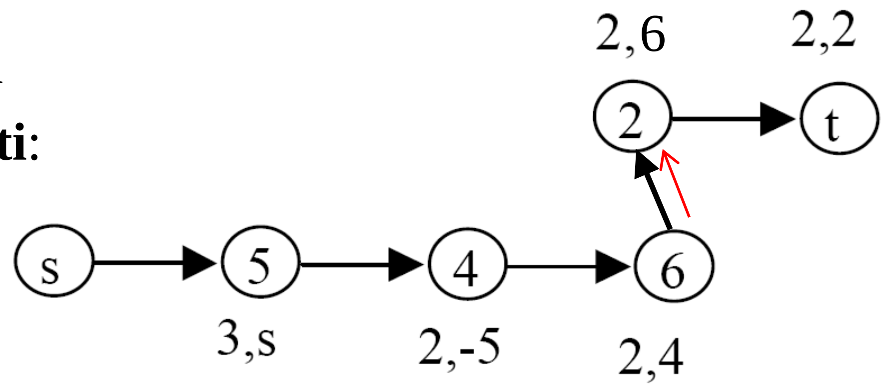


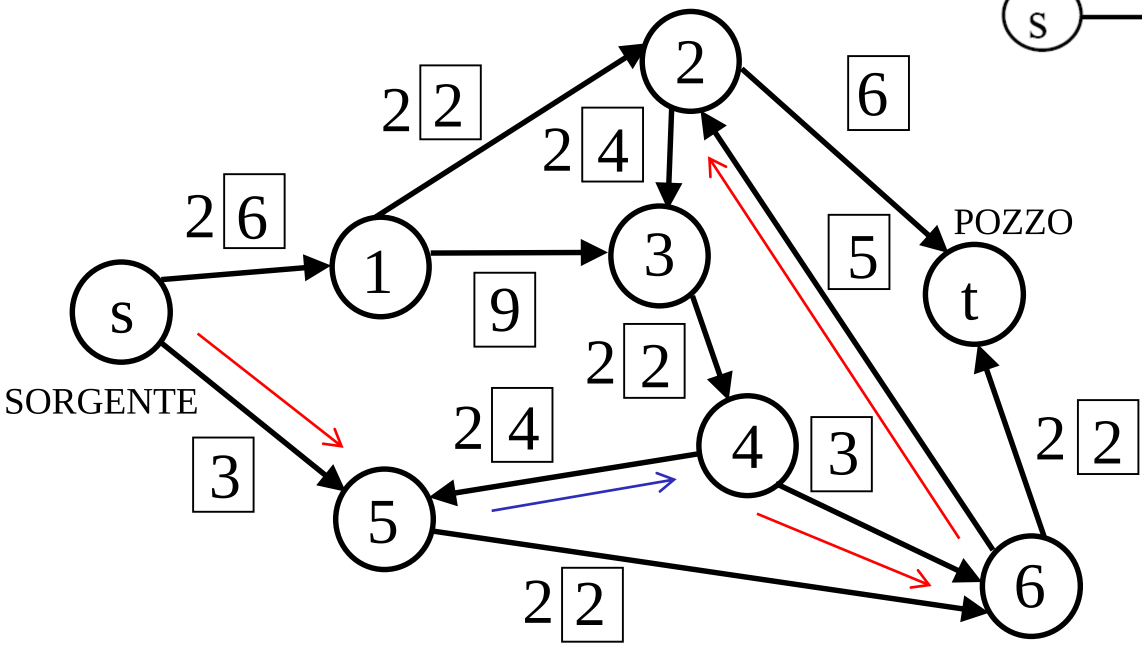


Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2

Esercizi Algo di Ford–Fulkerson (6)

La ricerca del I cammino aumentante porta a costruire un **albero dei cammini aumentanti**:

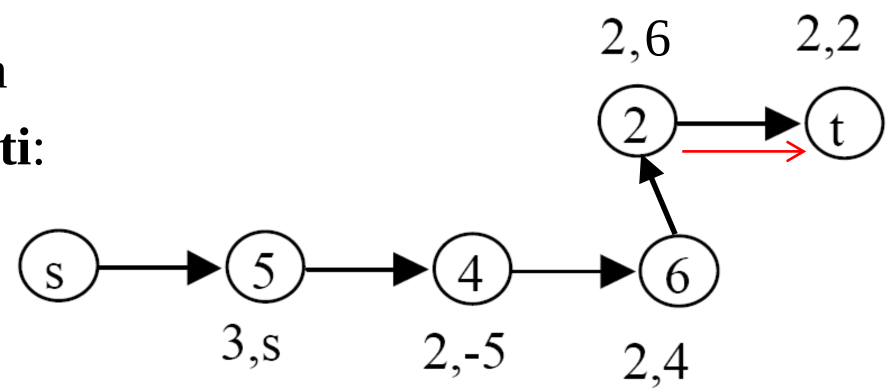


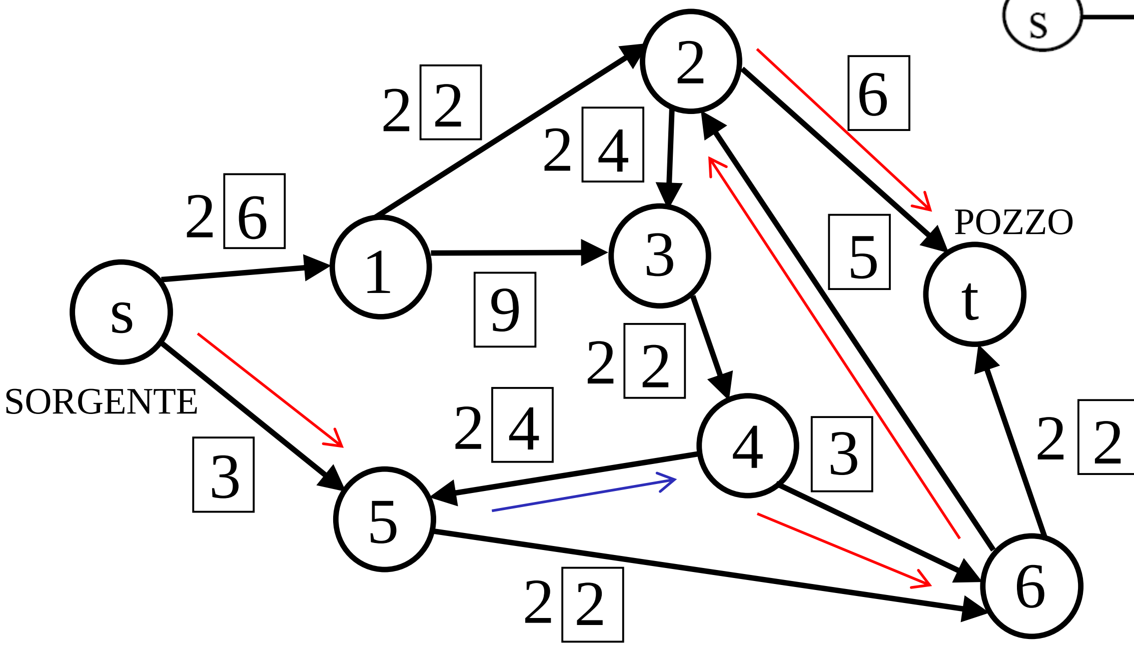


Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2

Esercizi Algo di Ford–Fulkerson (7)

La ricerca del I cammino aumentante porta a costruire un **albero dei cammini aumentanti**:

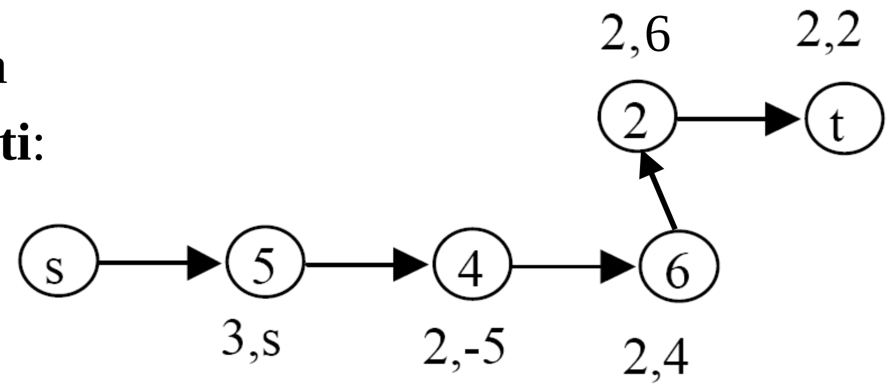


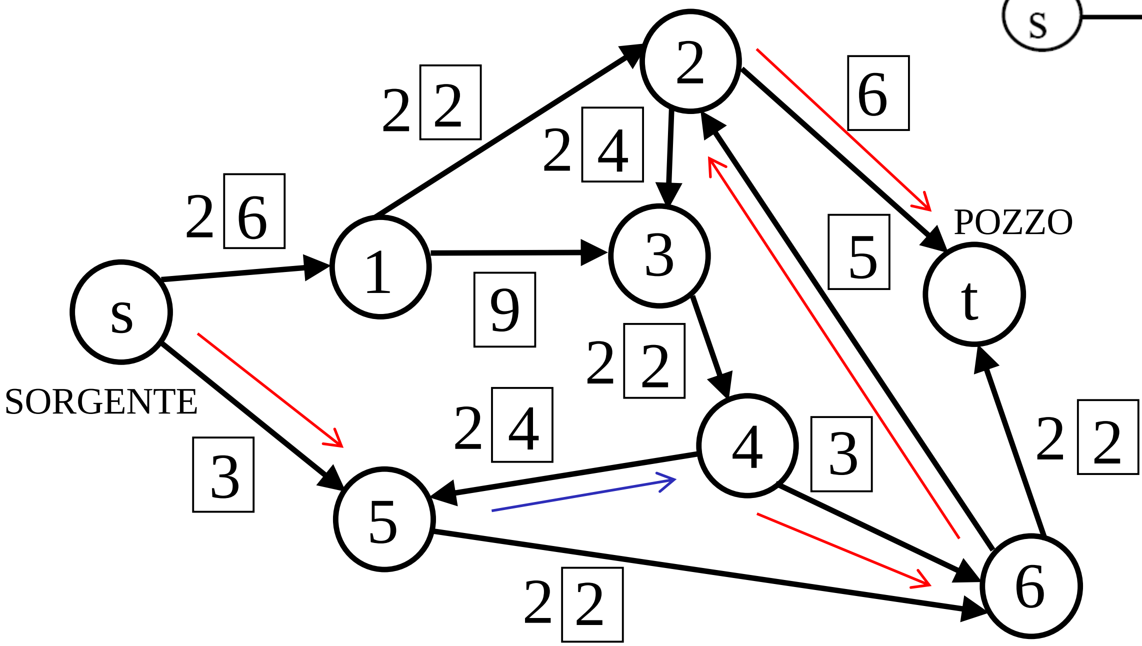


Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2

Esercizi Algo di Ford–Fulkerson (8)

La ricerca del I cammino aumentante porta a costruire un **albero dei cammini aumentanti**:



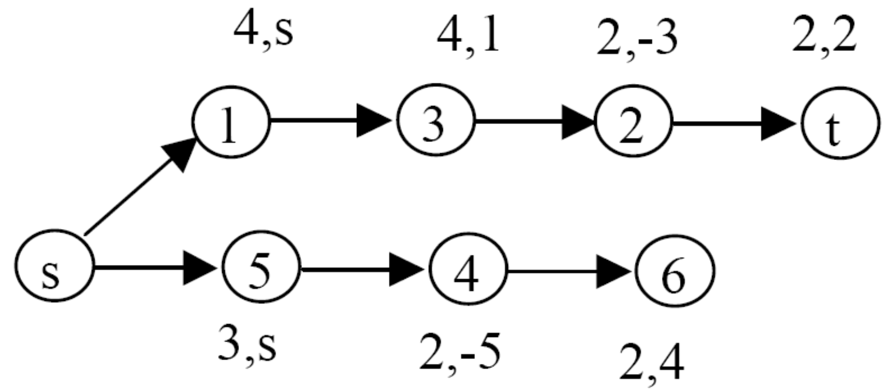


Oss: Il massimo flusso δ che posso trasportare da s a t è pari alla minima portata degli archi che attraversa

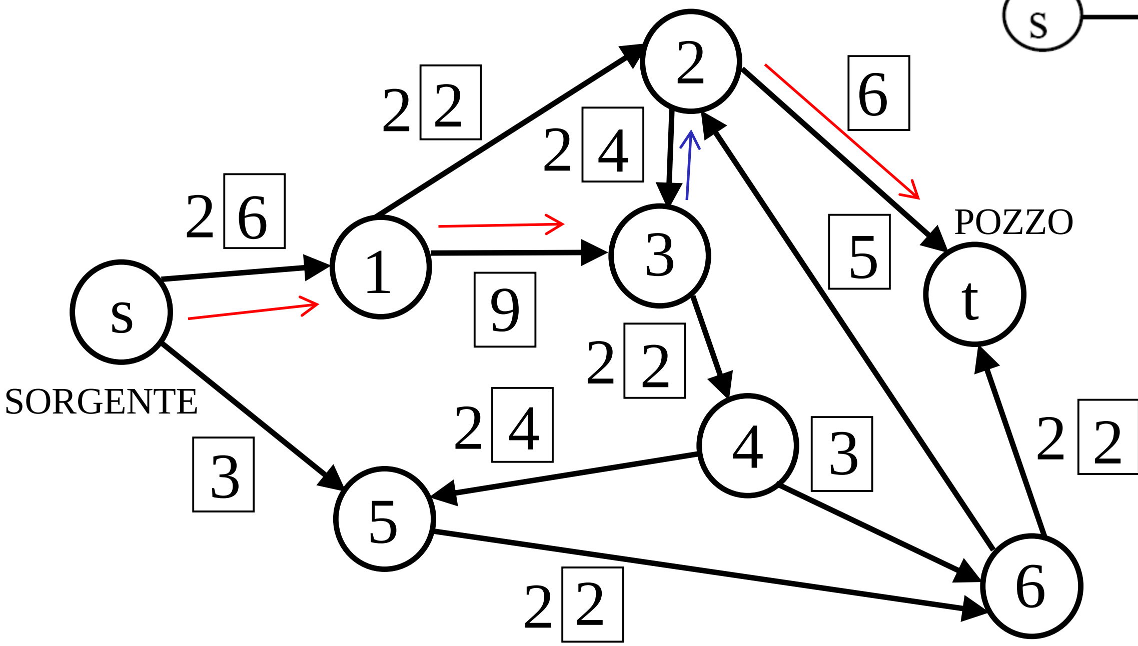
$$\boldsymbol{\delta = \min\{ r_{ij} : (i, j) \text{ in } P \} = 2}$$

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2
		+2				+2		-2	+2			+2	

Esercizi Algo di Ford–Fulkerson (9)



In alternativa si può ottenere il seguente **albero dei cammini aumentanti**:

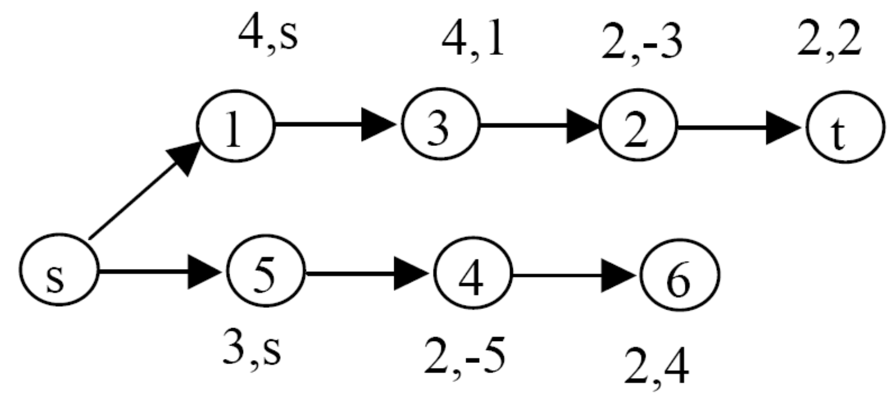


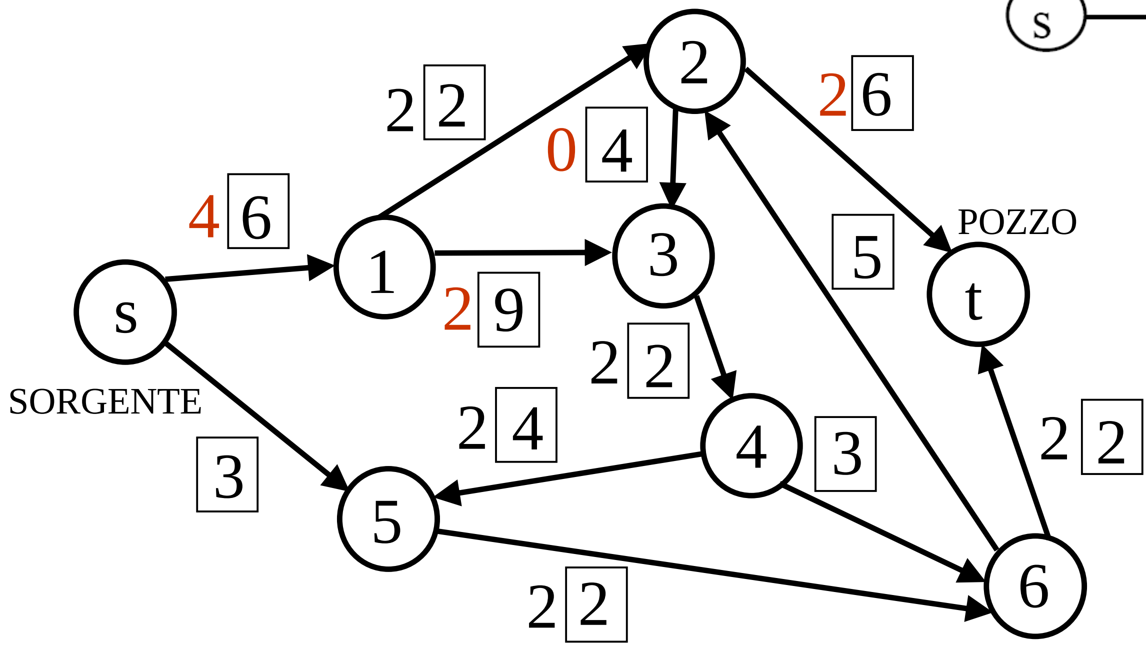
Oss: Posso portare un flusso pari a 2 unità sia dal nodo 3 che dal nodo 6 verso i nodi 2 e successivamente *t*

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2
	+2			+2	-2	+2							

Esercizi Algo di Ford–Fulkerson (10)

Sulla base del cammino aumentante trovato, aggiorno i flussi sulla rete:



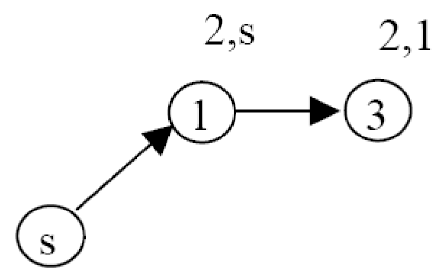


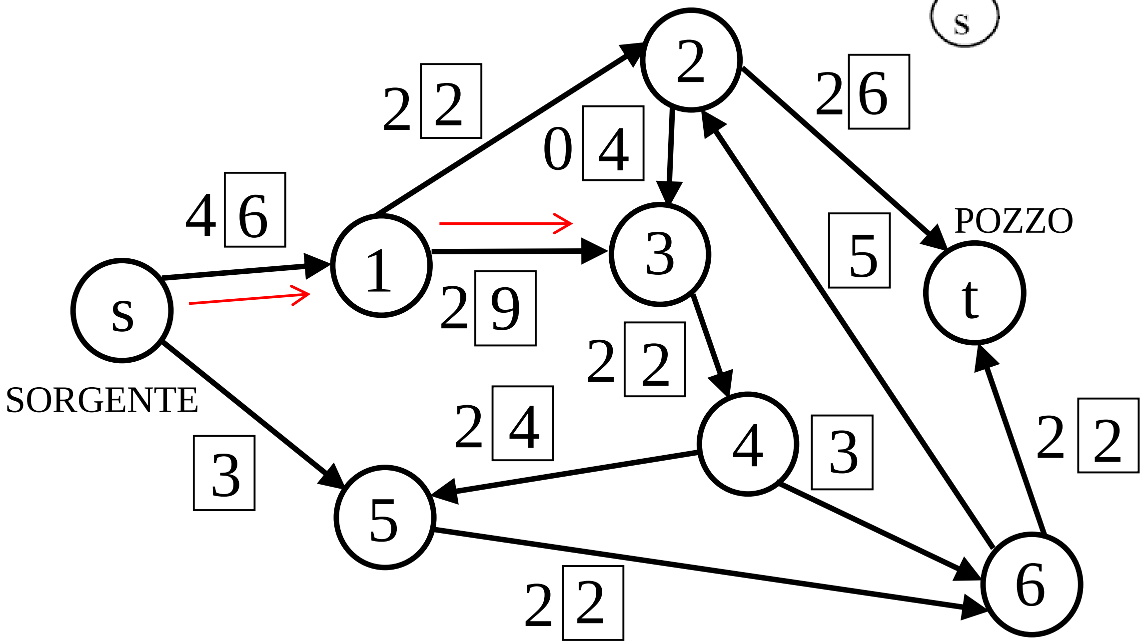
$\delta = 2 \;\; v = 4$

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2
	+2			+2	-2	+2							

Esercizi Algo di Ford–Fulkerson (11)

La ricerca del II cammino aumentante porta a costruire il seguente albero:



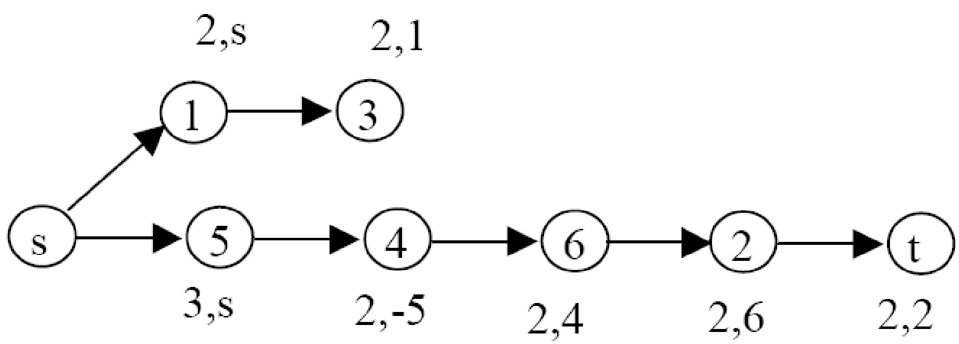


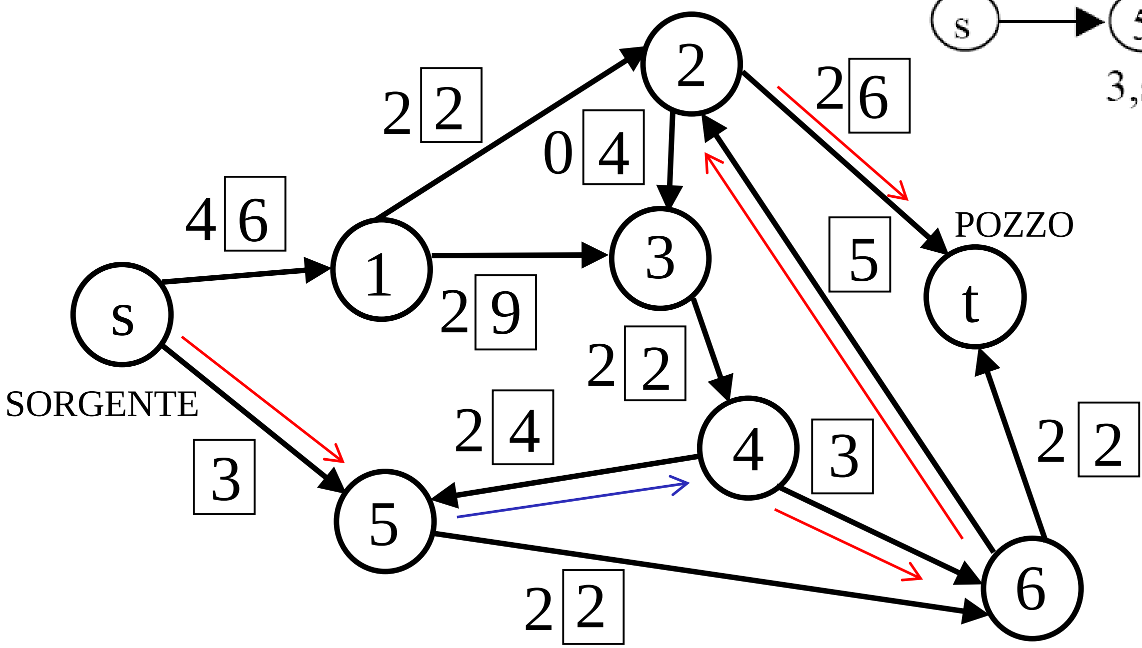
Oss: Dal nodo 3 non posso portare flusso dato che l'arco inverso (3,2) è già scarico e l'arco diretto (3,4) è saturo

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	4	0	2	2	0	2	2	2	0	0	2	0	2

Esercizi Algo di Ford–Fulkerson (12)

La ricerca del II cammino aumentante porta a costruire il seguente albero:





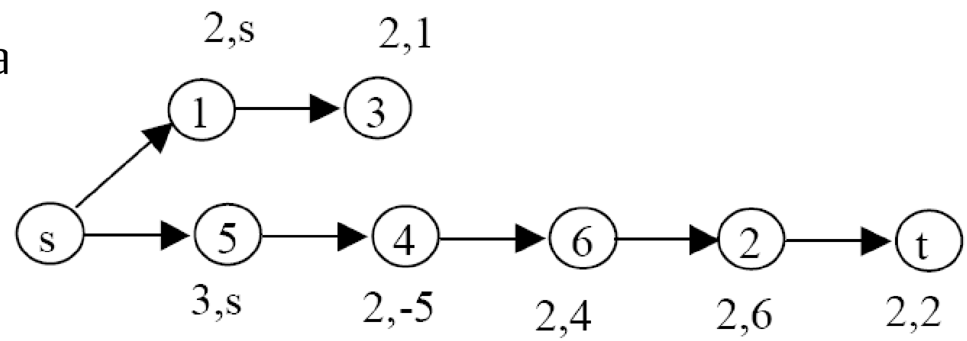
Oss: Il massimo flusso δ che posso trasportare da s a t è pari alla minima portata degli archi che attraversa

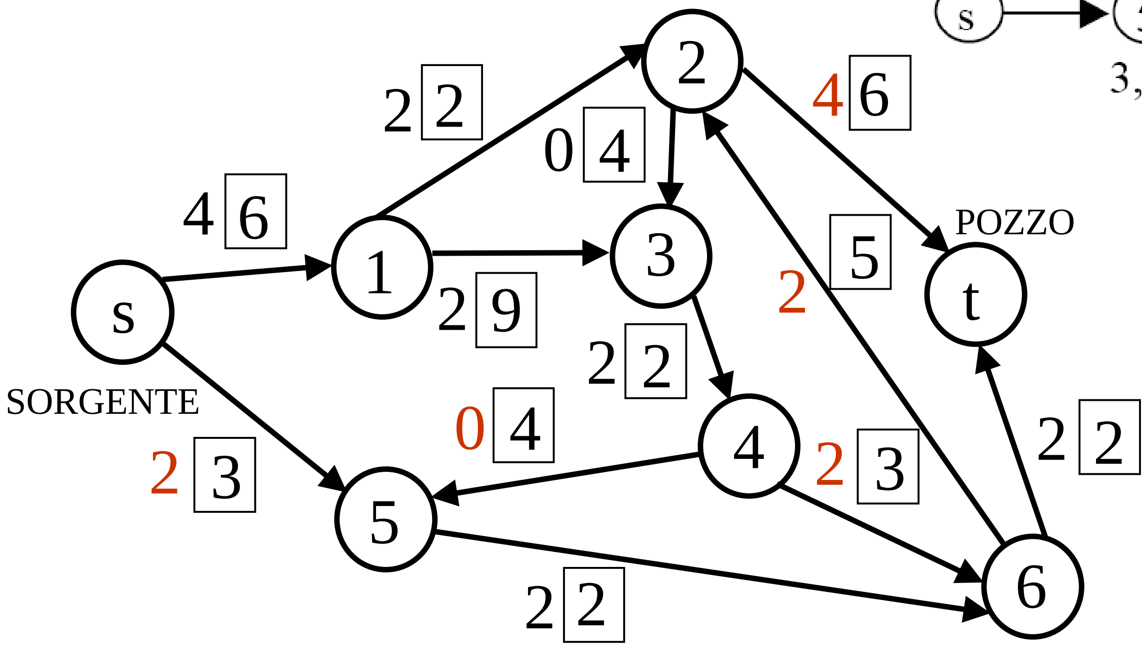
$\boldsymbol{\delta = \min\{ r_{ij} : (i, j) \text{ in } P \} = 2}$

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	4	0	2	2	0	2	2	2	0	0	2	0	2
		+2				+2		-2	+2			+2	

Esercizi Algo di Ford–Fulkerson (13)

Sulla base del cammino aumentante trova aggiorno i flussi sulla rete:



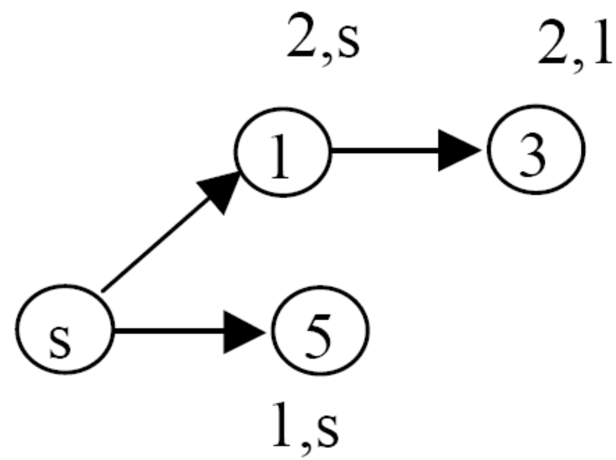


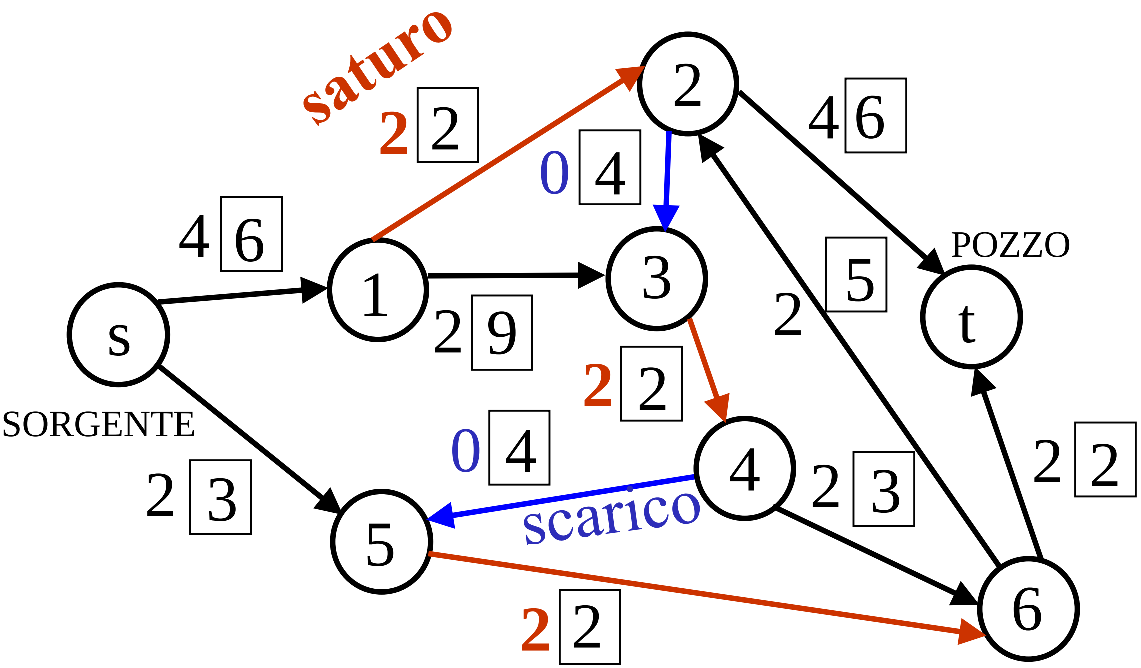
$\delta = 2 \quad v = 6$

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	4	0	2	2	0	2	2	2	0	0	2	0	2
		+2				+2		-2	+2			+2	

Esercizi Algo di Ford–Fulkerson (14)

La ricerca del III cammino aumentante porta a costruire il seguente albero:





S = [s,1,3,5], $\underline{S}$ = [2,4,6,t]
ho un taglio di capacità 6, pari alla capacità degli archi uscenti (archi rossi) dal taglio: (1,2),(3,4),(5,6)
[archi blu = archi entranti]

Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	4	2	2	2	0	4	2	0	2	0	2	2	2

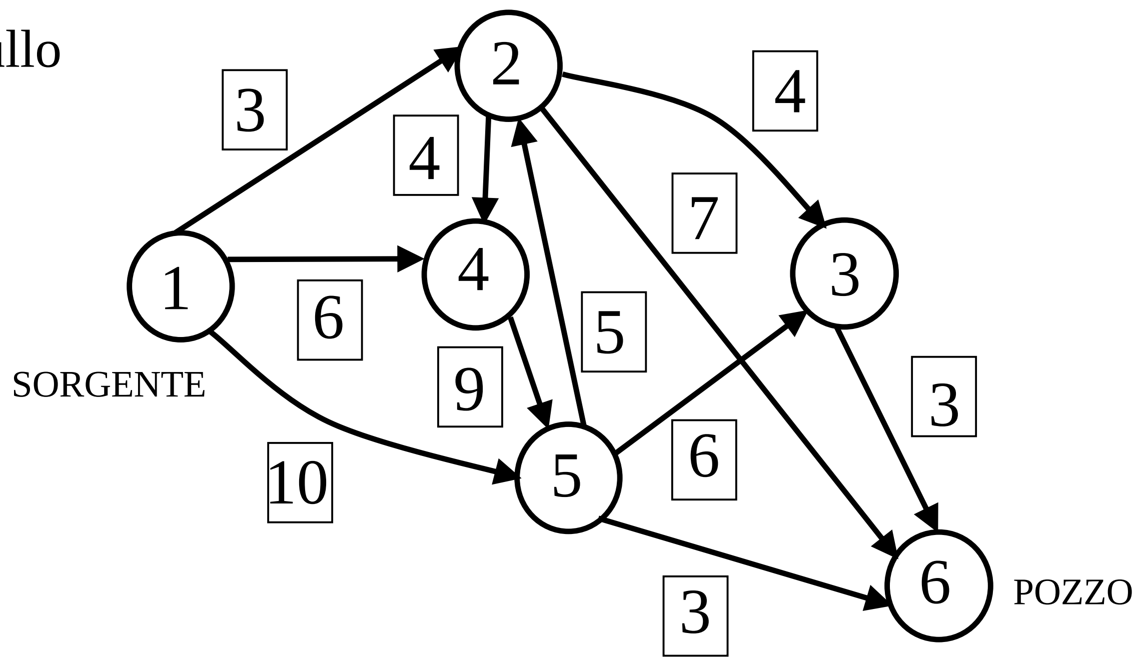
Esercizi Algo di Ford–Fulkerson (15)

Es. 2: Nella tabella sono riportate le caratteristiche degli 11 archi di un digrafo con 6 nodi. Si individui il taglio di capacità minima tra i nodi 1 (*s*) e 6 (*t*) del digrafo, specificando la capacità del taglio e gli archi che lo compongono.

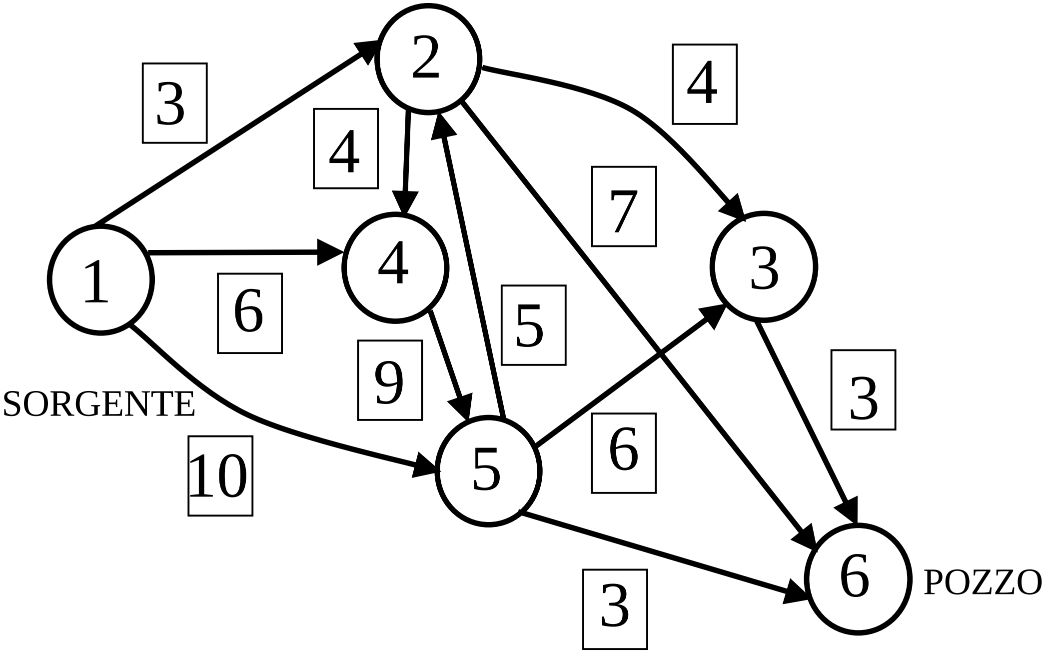
Archi	(1,2)	(1,4)	(1,5)	(2,3)	(2,4)	(2,6)	(3,6)	(4,5)	(5,2)	(5,3)	(5,6)
Capacità	3	6	10	4	4	7	3	9	5	6	3

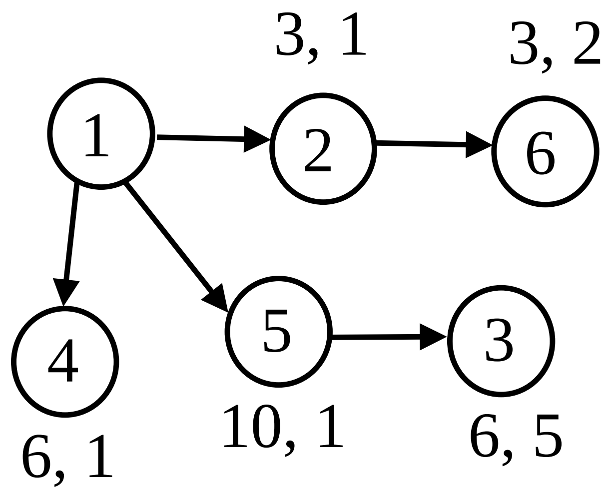
Il flusso iniziale è nullo

$\delta = 0 \quad v = 0$



Esercizi Algo di Ford–Fulkerson (16)



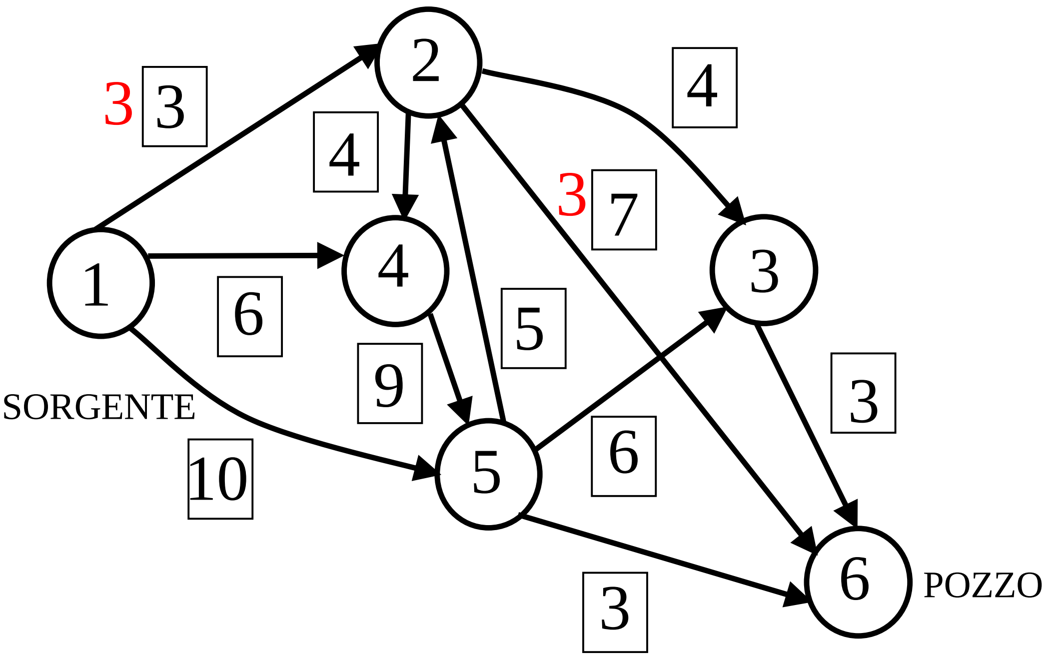


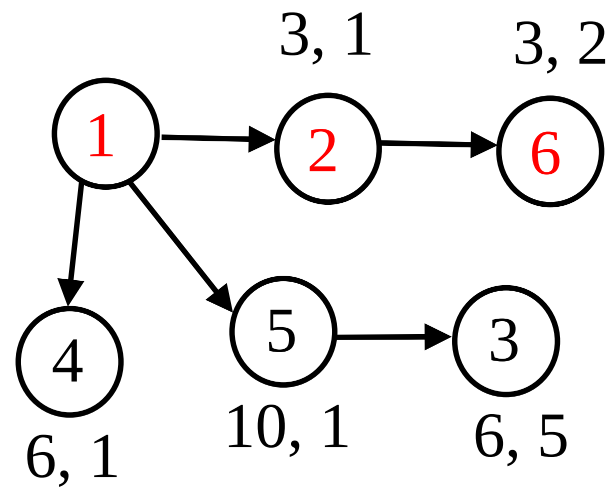
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$

Esercizi Algo di Ford–Fulkerson (17)



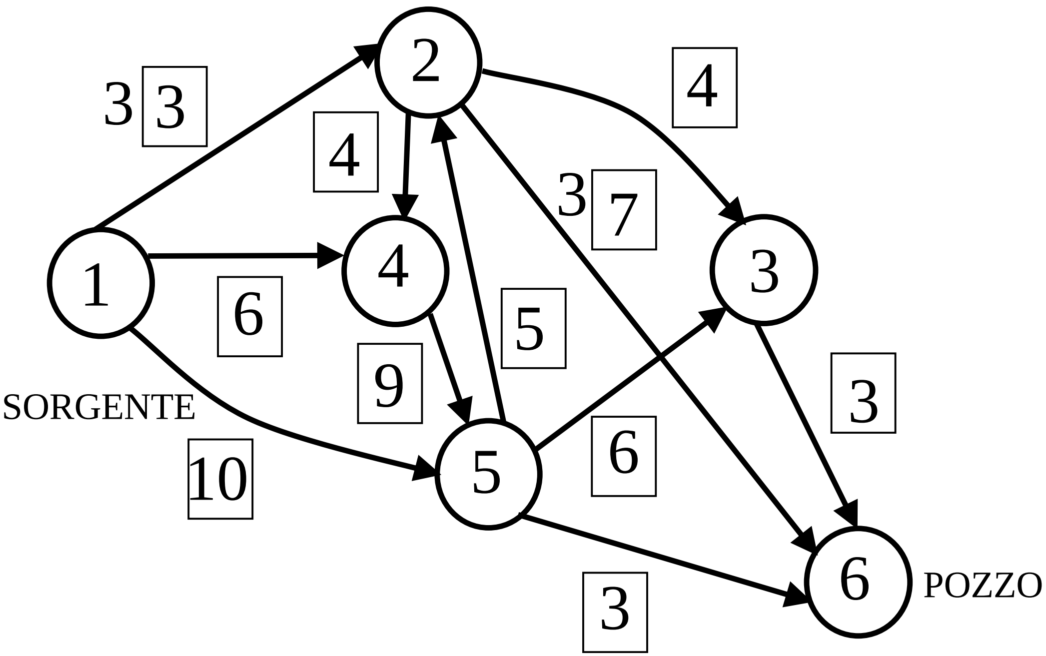


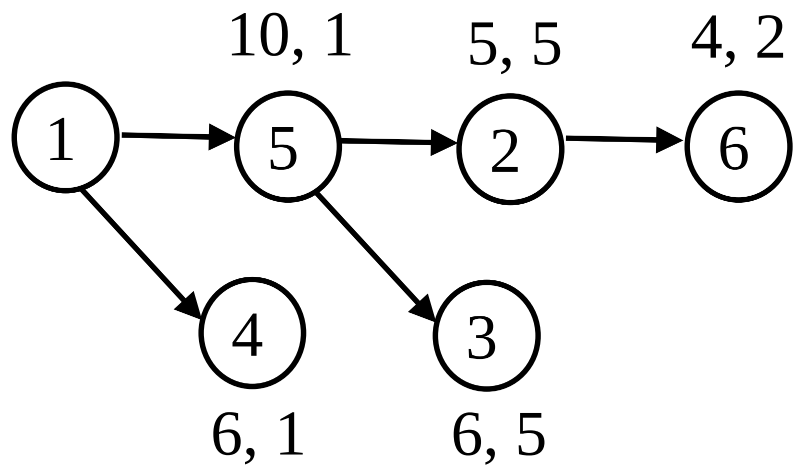
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$

Esercizi Algo di Ford–Fulkerson (18)



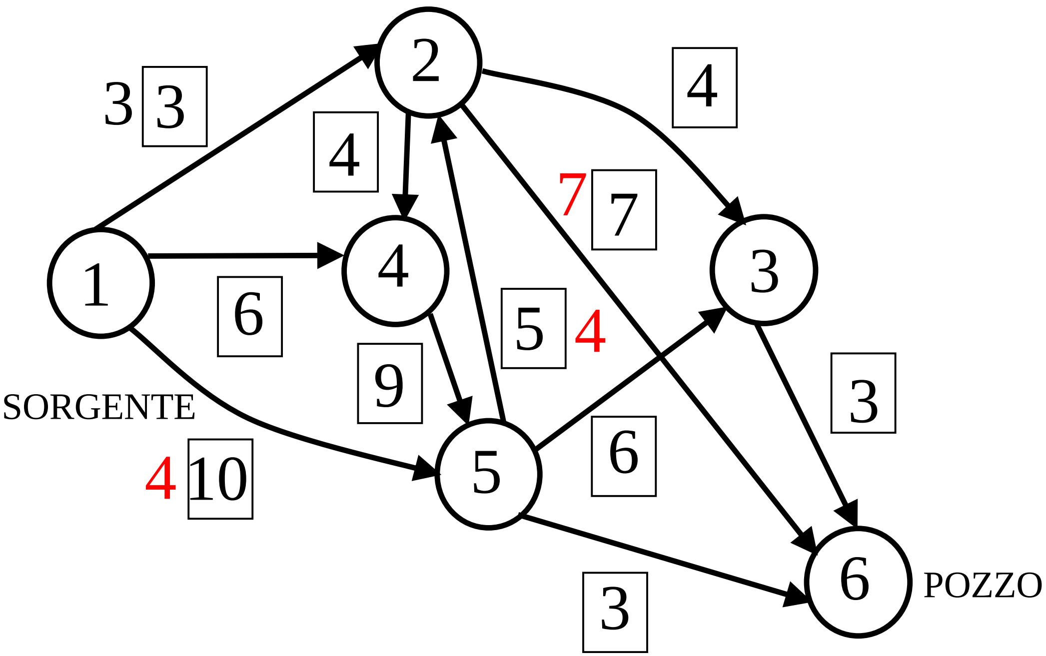


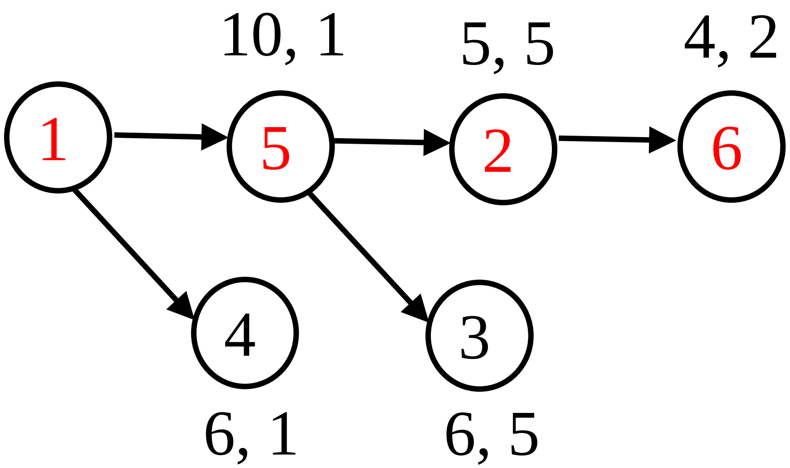
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$ 1-5-2-6 $\delta = 4$ $v = 7$

Esercizi Algo di Ford–Fulkerson (19)



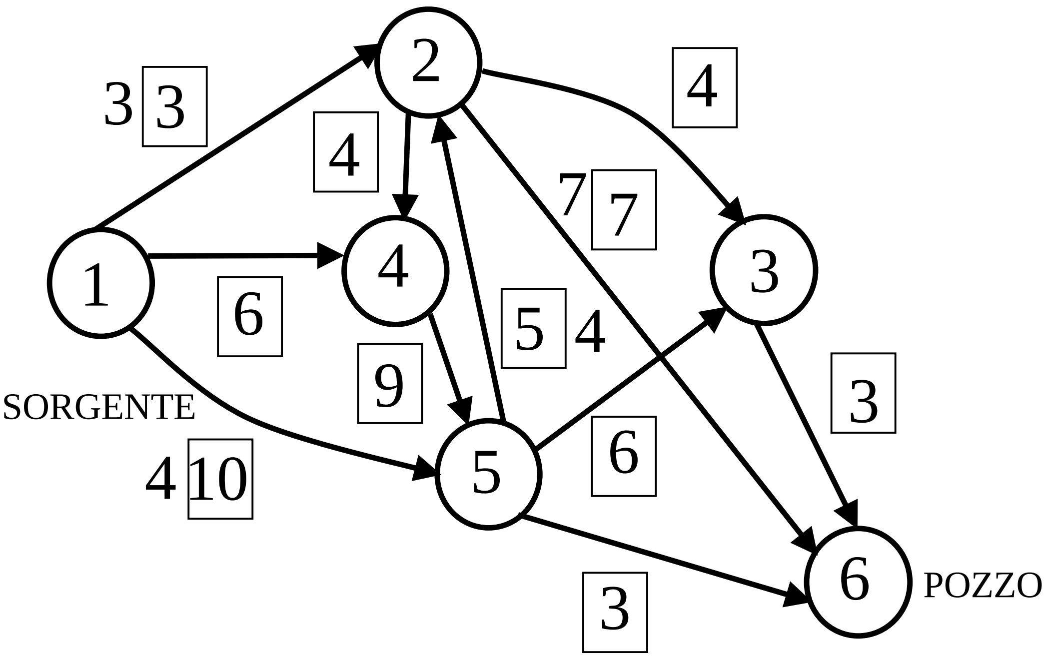


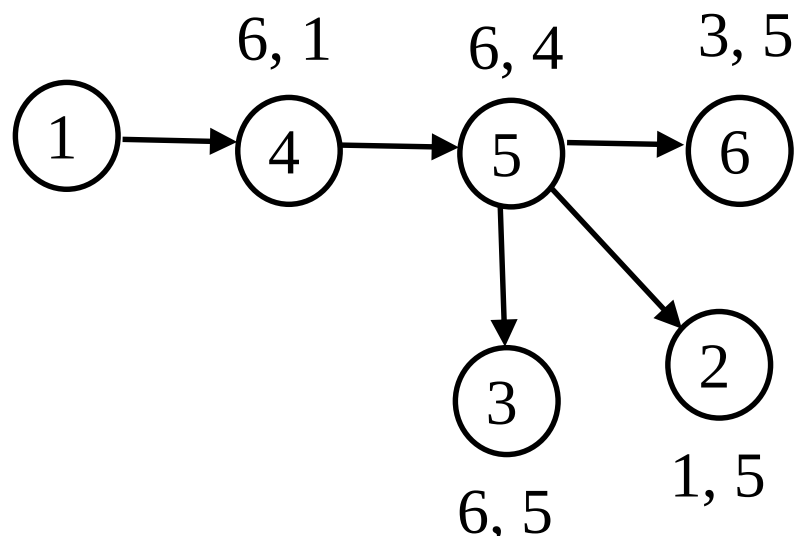
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$ 1-5-2-6 $\delta = 4$ $v = 7$

Esercizi Algo di Ford–Fulkerson (20)





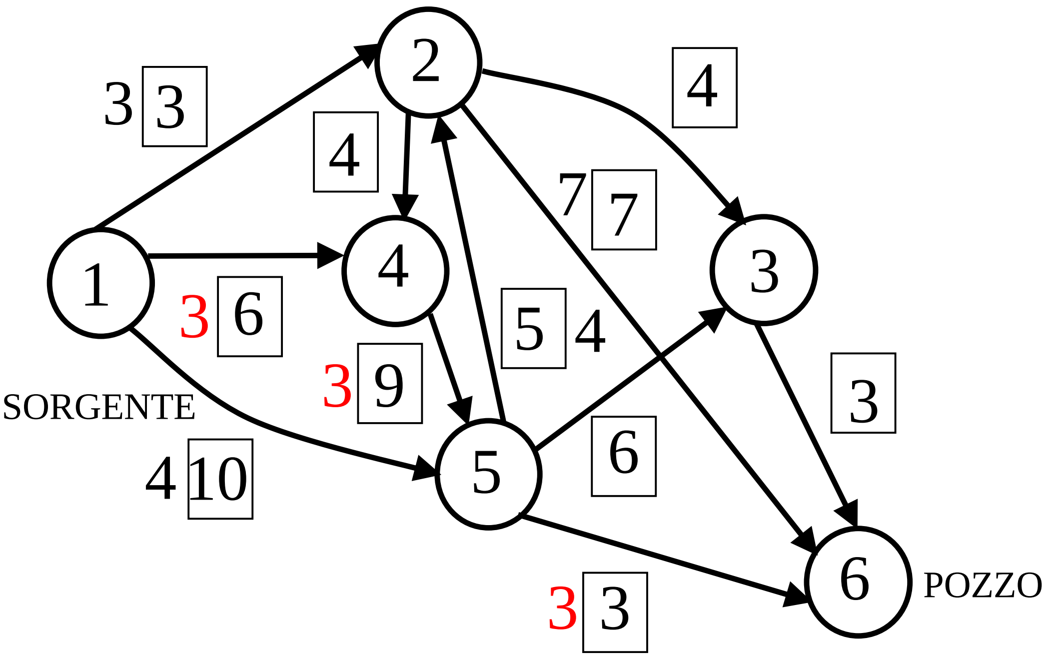
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

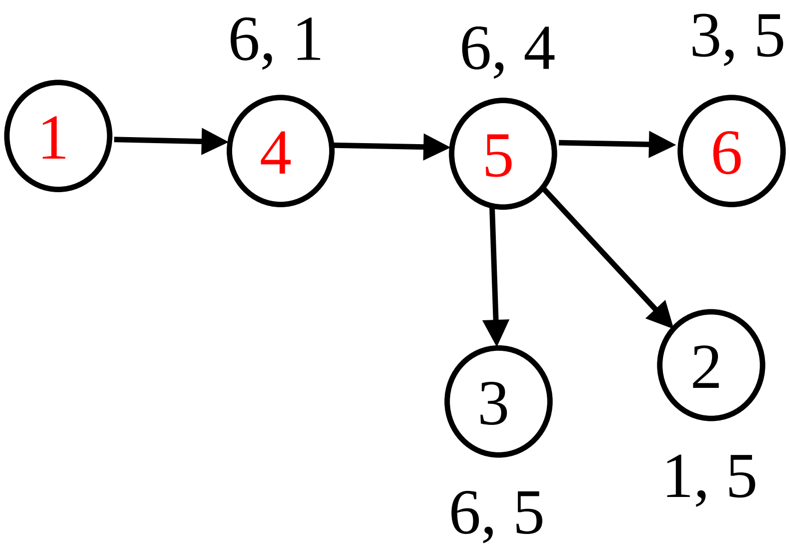
Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$ 1-5-2-6 $\delta = 4$ $v = 7$

1-4-5-6 $\delta = 3$ $v = 10$

Esercizi Algo di Ford–Fulkerson (21)





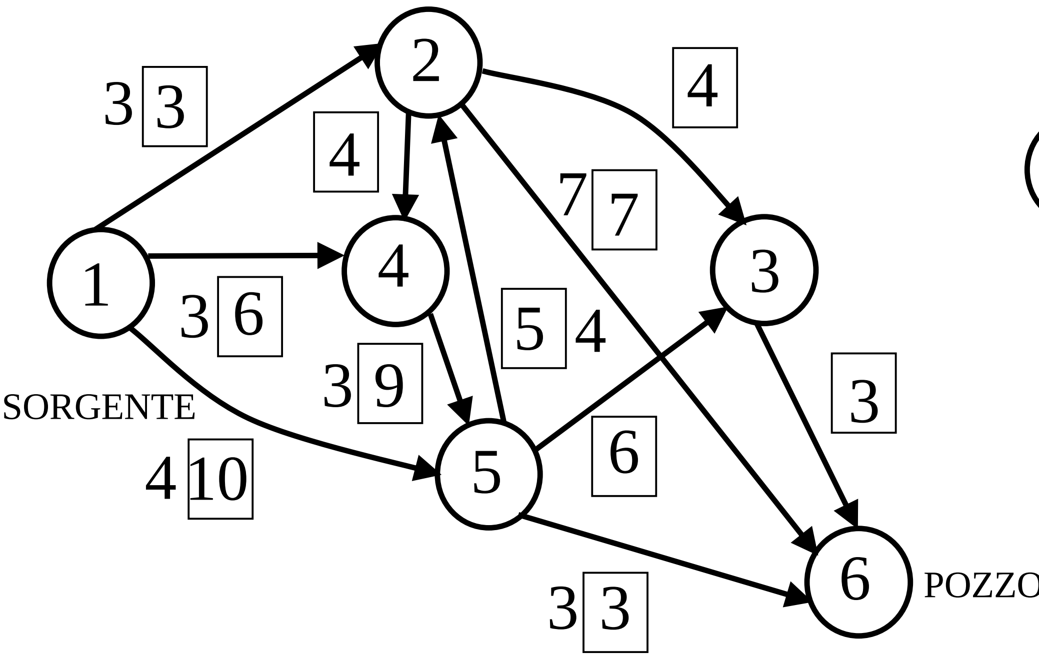
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

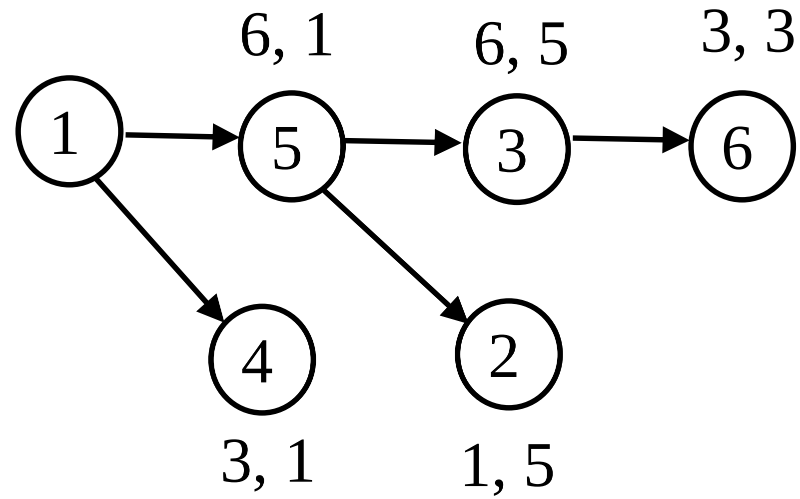
Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$ 1-5-2-6 $\delta = 4$ $v = 7$

1-4-5-6 $\delta = 3$ $v = 10$

Esercizi Algo di Ford–Fulkerson (22)





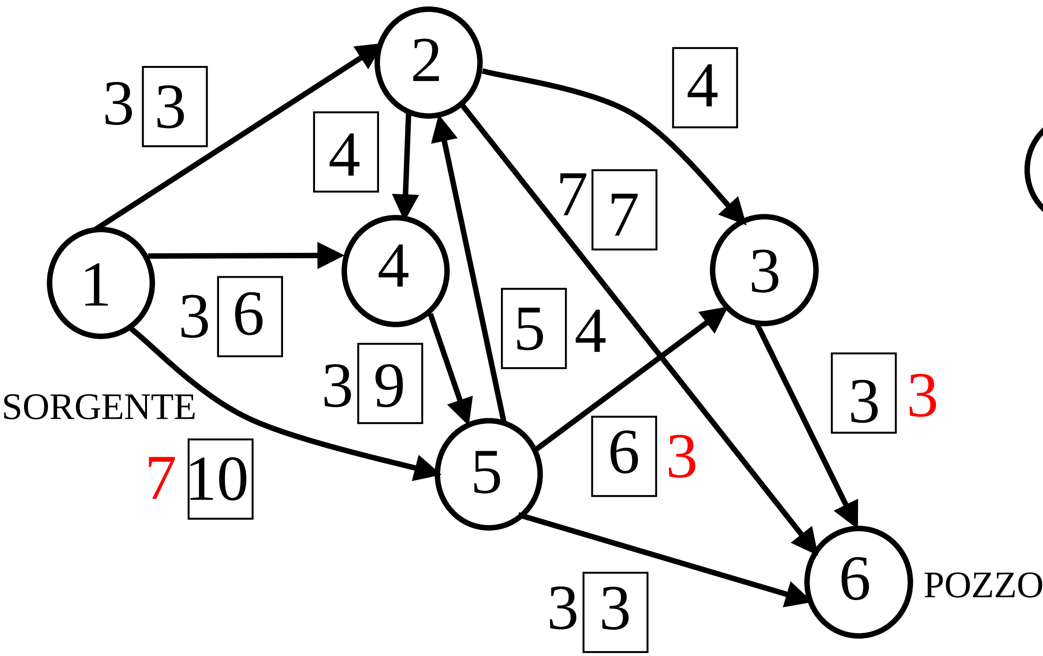
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

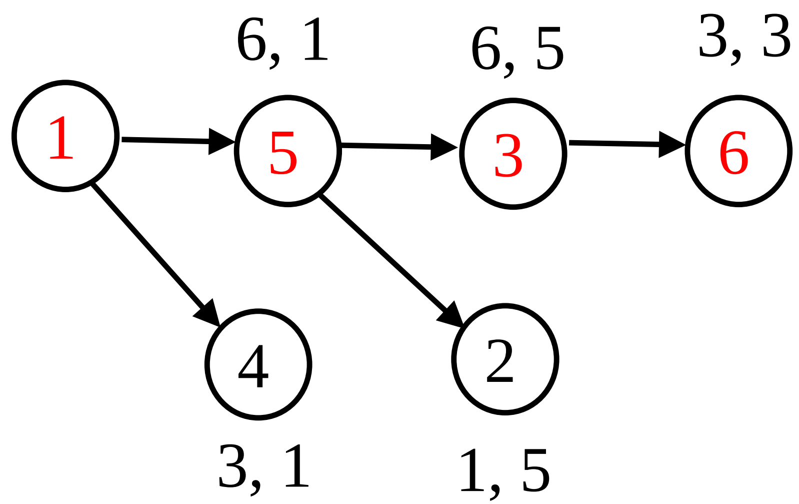
Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$ 1-5-2-6 $\delta = 4$ $v = 7$

1-4-5-6 $\delta = 3$ $v = 10$ 1-5-3-6 $\delta = 3$ $v = 13$

Esercizi Algo di Ford–Fulkerson (23)





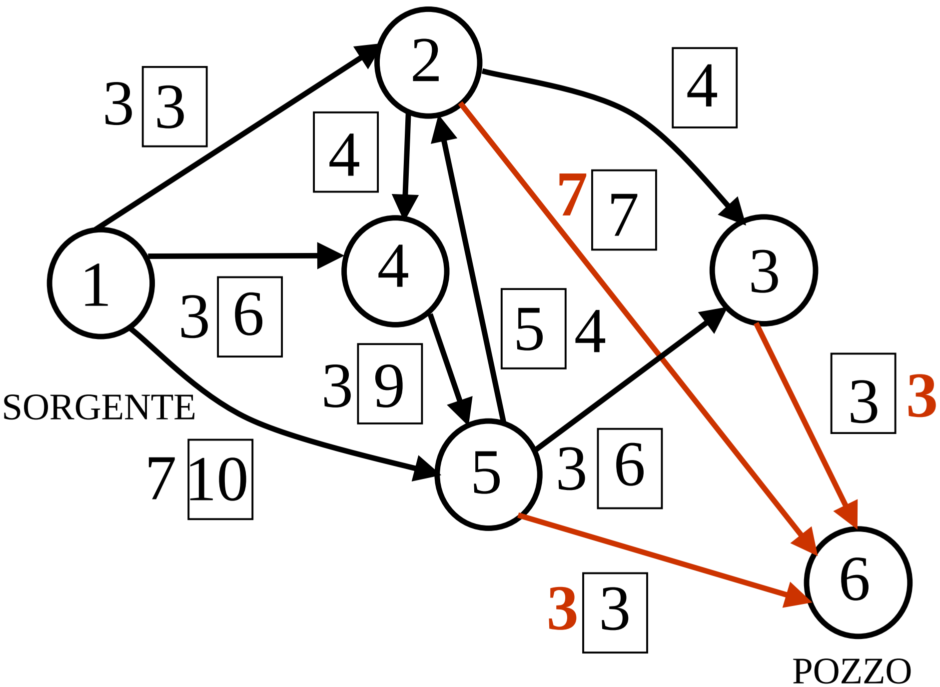
Il taglio di capacità minima si calcola stabilendo il massimo flusso nel digrafo.

Ford-Fulkerson determina <u>una possibile scelta</u> dei cammini aumentanti:

1-2-6 $\delta = 3$ $v = 3$ 1-5-2-6 $\delta = 4$ $v = 7$

1-4-5-6 $\delta = 3$ $v = 10$ 1-5-3-6 $\delta = 3$ $v = 13$

Esercizi Algo di Ford–Fulkerson (24)



$v = 13$

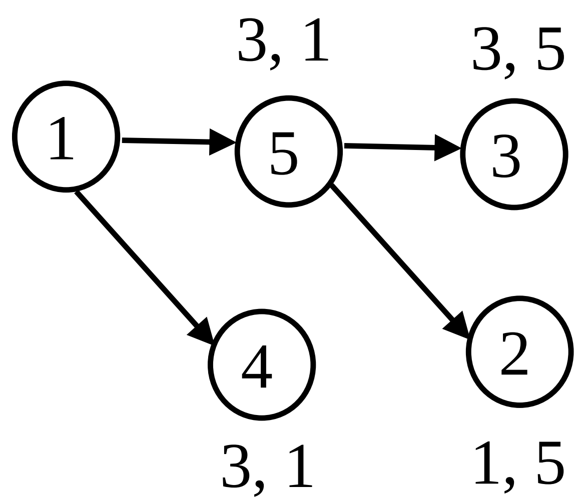
Il taglio di capacità minima è

***S* = [1,2,3,4,5], <u>*S*</u> = [6],**

gli archi che lo compongono

sono (2,6), (3,6) e (5,6).

La capacità del taglio è **13**.



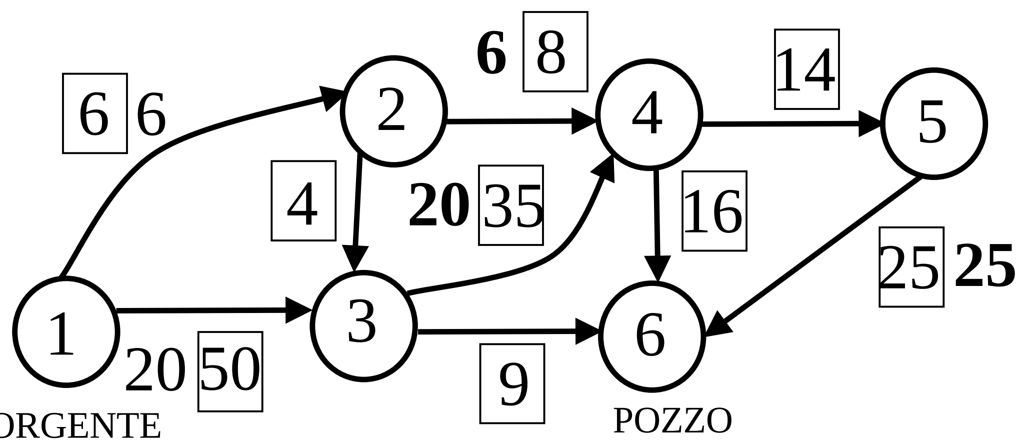
Esercizi Algo di Ford–Fulkerson (25)

Es. 3: In tabella sono riportati gli archi di un digrafo con 6 nodi, e sono dati i valori di capacità degli archi ed un flusso iniziale. Domande:

(1) Si verifichi che la distribuzione di flusso iniziale sia ammissibile
(2) Se il flusso dato è ammissibile allora passare ai punti 4-5
(3) Se il flusso non è ammissibile partire dal digrafo completamente scarico
(4) Trovare il massimo flusso inviabile da 1 a 6 con l'algo di Ford e Fulkerson
(5) Individuare il taglio di capacità minima nel digrafo

Archi	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	(3,6)	(4,5)	(4,6)	(5,6)
Capacità	6	50	4	8	35	9	14	16	25
Flussi	6	20	0	6	20	0	0	0	25

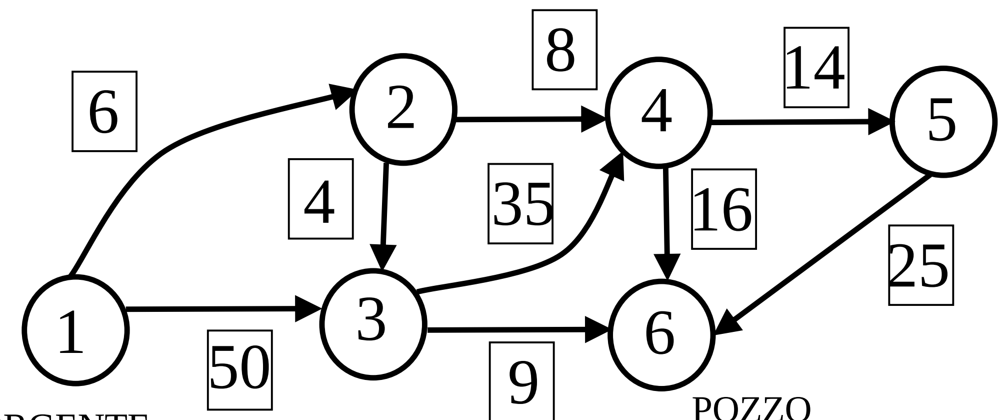
Il flusso iniziale <u>non è ammissibile</u> in quanto non rispetta alcuni vincoli di bilanciamento delle masse:
sul nodo 4 (26 entrante, 0 uscente)
sul nodo 5 (0 entrante, 25 uscente)



Esercizi Algo di Ford–Fulkerson (26)

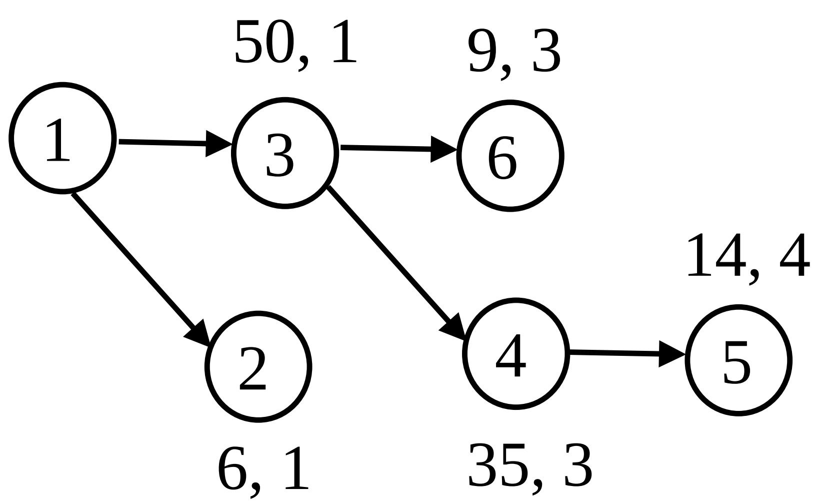
(1) Il flusso dato non è sia ammissibile (violati i vincoli di bilanciamento)

(3) Da cui, partiamo dal digrafo completamente scarico (rete vuota):



(4) I cammini aumentanti sono:

1,3,6 $\delta = 9$ $v = 9$



<u>Osservazione 1</u>: C'era un cammino aumentante con maggior flusso (14) da 1 a 6 passante per i nodi 3, 4, 5.

<u>Osservazione 2</u>: Scelto un cammino aumentante, bisogna prendere il massimo flusso che vi scorre.

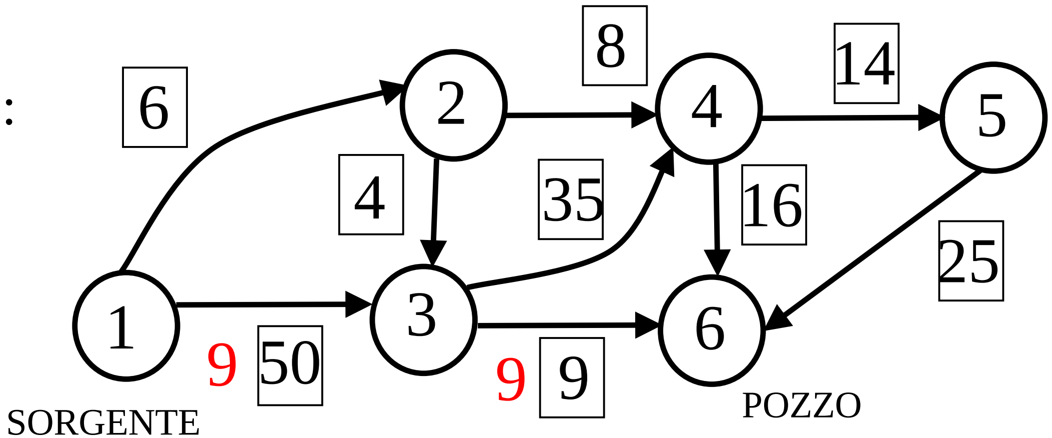
Esercizi Algo di Ford–Fulkerson (27)

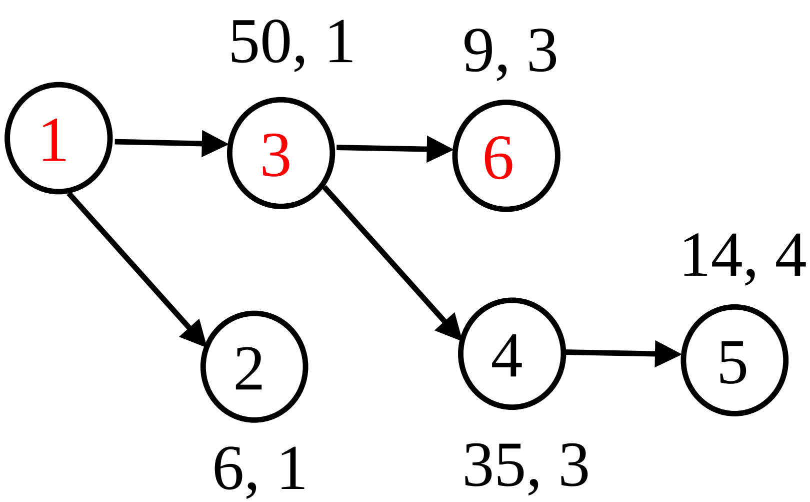
(1) Il flusso dato non è sia ammissibile (violati i vincoli di bilanciamento)

(3) Da cui, partiamo dal digrafo completamente scarico (rete vuota):

(4) I cammini aumentanti sono:

1,3,6 $\boldsymbol{\delta = 9}$ $\boldsymbol{v = 9}$

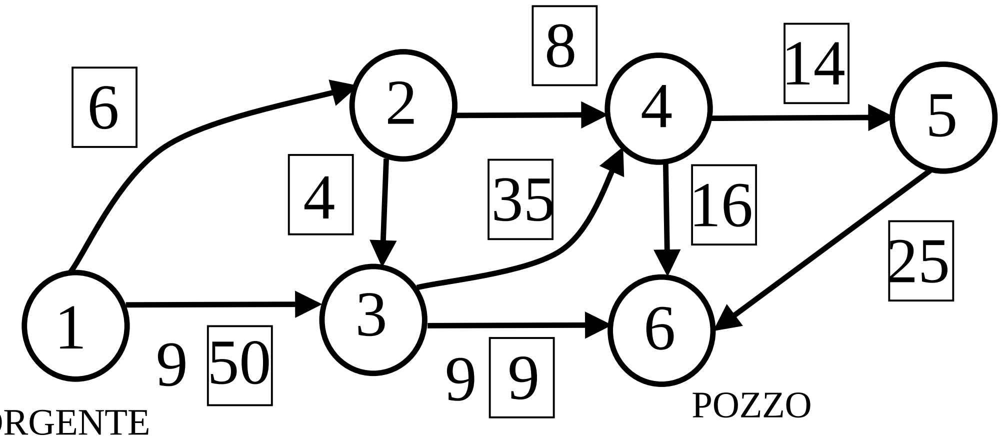




Esercizi Algo di Ford–Fulkerson (28)

(1) Il flusso dato non è sia ammissibile (violati i vincoli di bilanciamento)

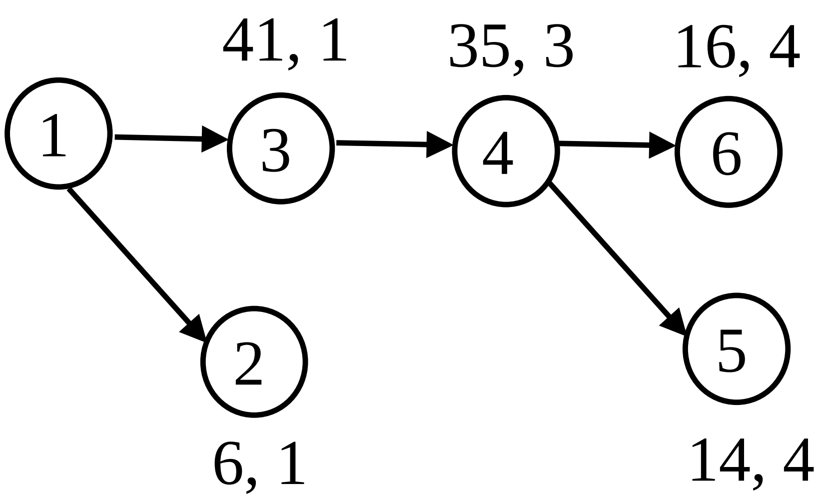
(3) Da cui, partiamo dal digrafo completamente scarico (rete vuota):



(4) I cammini aumentanti sono:

1,3,6 $\delta = 9$ $v = 9$

1,3,4,6 $\boldsymbol{\delta = 16}$ $\boldsymbol{v = 25}$



Esercizi Algo di Ford–Fulkerson (29)

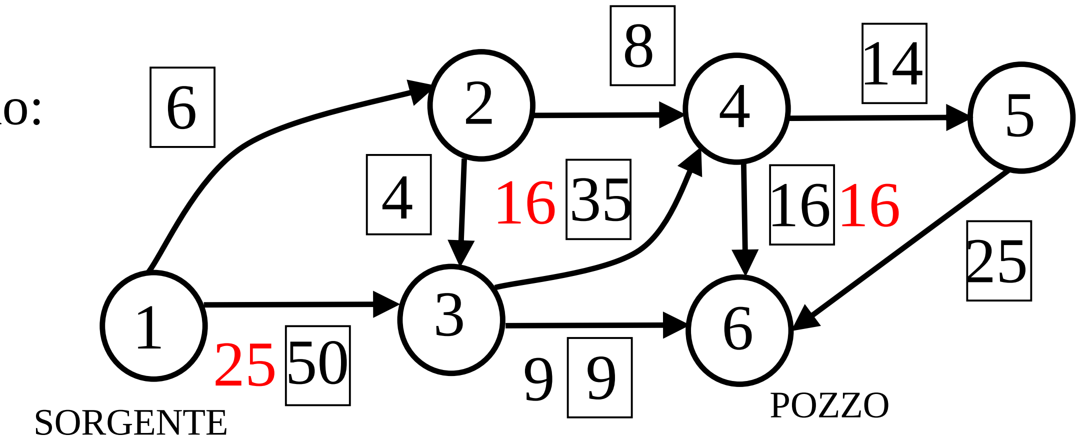
(1) Il flusso dato non è sia ammissibile (violati i vincoli di bilanciamento)

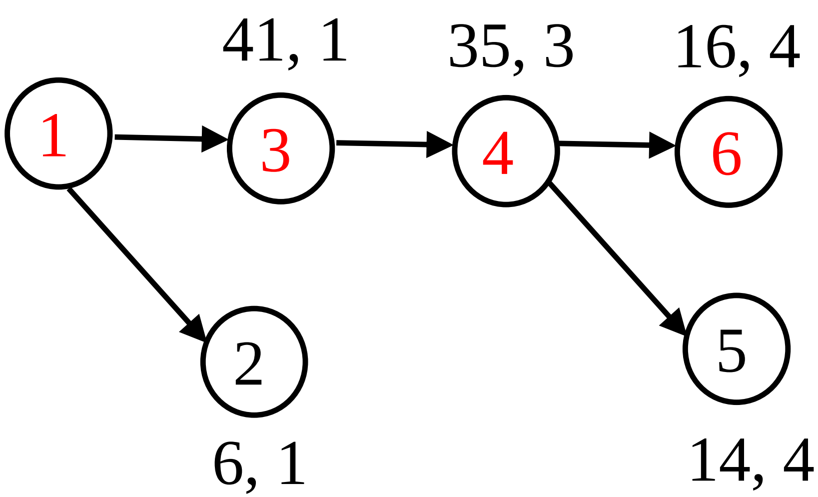
(3) Da cui, partiamo dal digrafo completamente scarico (rete vuota):

(4) I cammini aumentanti sono:

1,3,6 $\delta = 9$ $v = 9$

1,3,4,6 $\delta = 16$ $v = 25$





Esercizi Algo di Ford–Fulkerson (30)

(1) Il flusso dato non è sia ammissibile (violati i vincoli di bilanciamento)

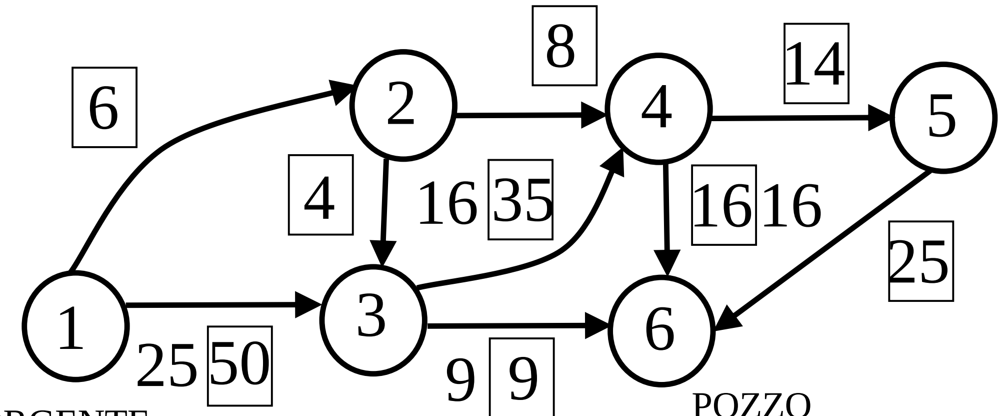
(3) Da cui, partiamo dal digrafo completamente scarico (rete vuota):

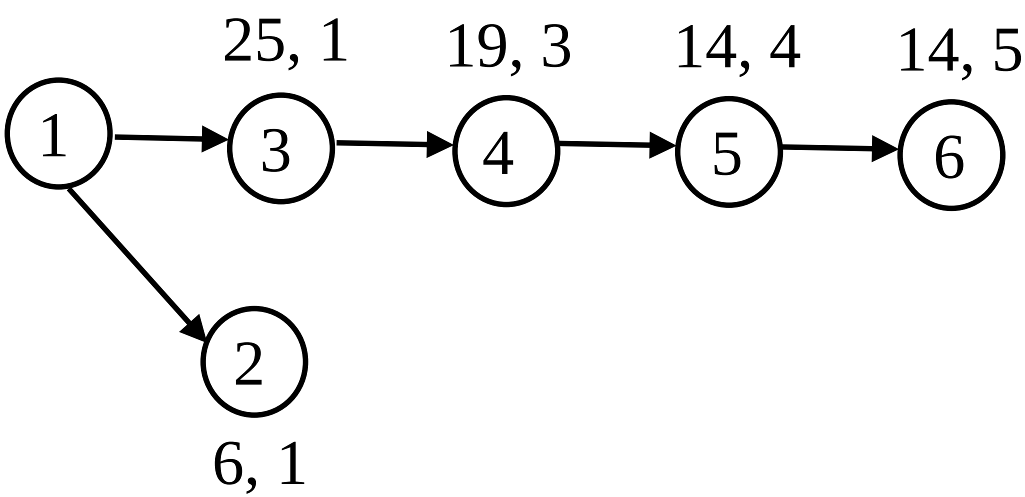
(4) I cammini aumentanti sono:

1,3,6 $\delta = 9$ $v = 9$

1,3,4,6 $\delta = 16$ $v = 25$

1,3,4,5,6 $\delta = 14$ $v = 39$





Esercizi Algo di Ford–Fulkerson (31)

(1) Il flusso dato non è sia ammissibile (violati i vincoli di bilanciamento)

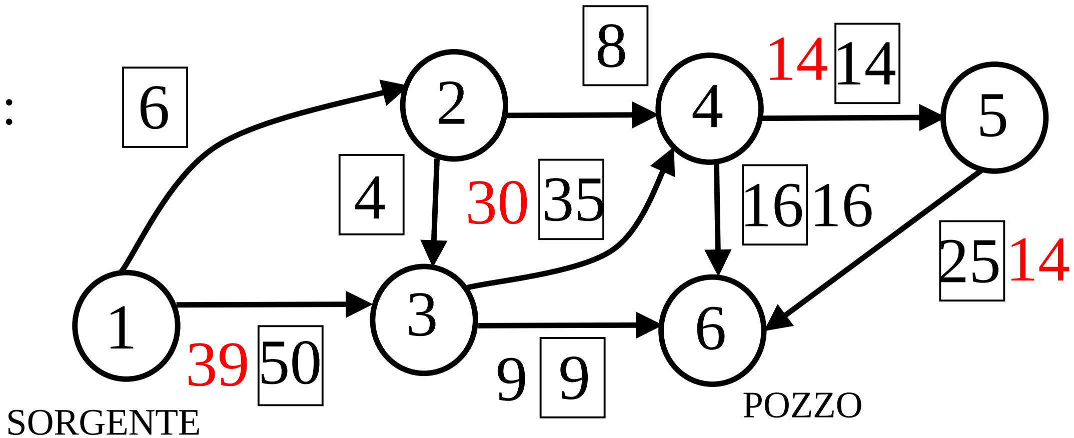
(3) Da cui, partiamo dal digrafo completamente scarico (rete vuota):

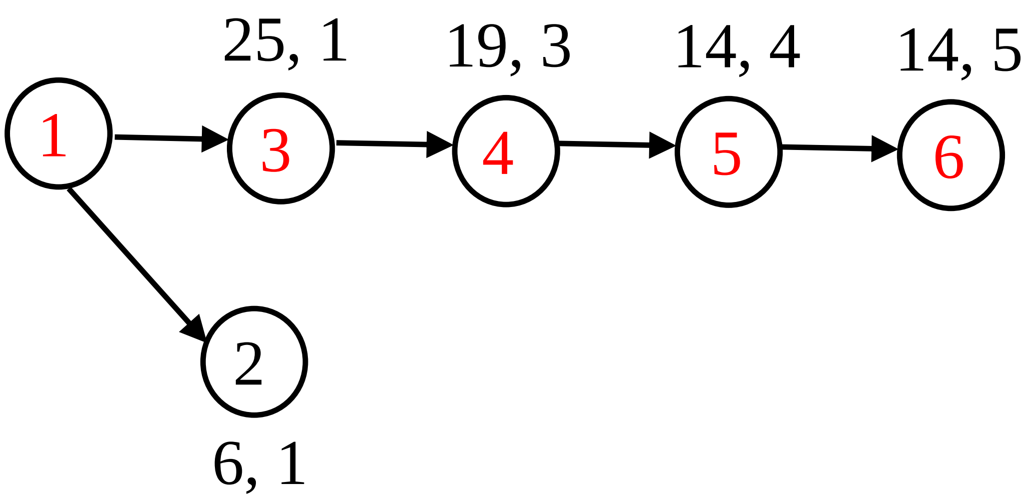
(4) I cammini aumentanti sono:

1,3,6 $\delta = 9$ $v = 9$

1,3,4,6 $\delta = 16$ $v = 25$

1,3,4,5,6 $\boldsymbol{\delta = 14}$ $\boldsymbol{v = 39}$





Esercizi Algo di Ford–Fulkerson (32)

(1) Il flusso dato non è sia ammissibile (violati i vincoli di bilanciamento)

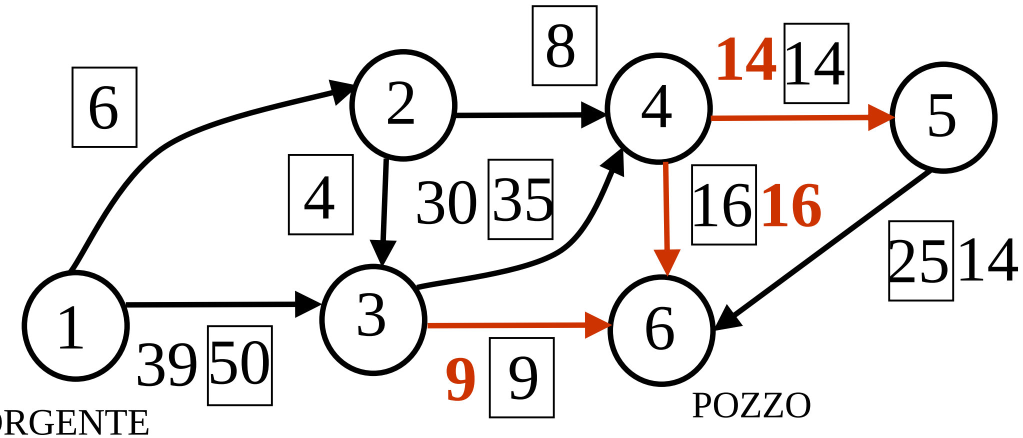
(3) Da cui, partiamo dal digrafo completamente scarico (rete vuota):

(4) I cammini aumentanti sono:

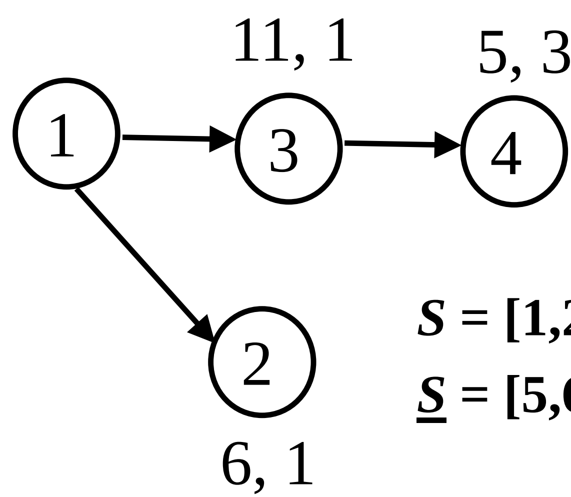
1,3,6 $\delta = 9$ $v = 9$

1,3,4,6 $\delta = 16$ $v = 25$

1,3,4,5,6 $\delta = 14$ $v = 39$



(5) Il taglio di costo minimo (di capacità minima) comprende i nodi 1, 2, 3, 4 con capacità pari a 39 => ottimo dato che $v = 39$



***S* = [1,2,3,4]**

***S̲* = [5,6]**

Esercizi Algo di Ford–Fulkerson (33)

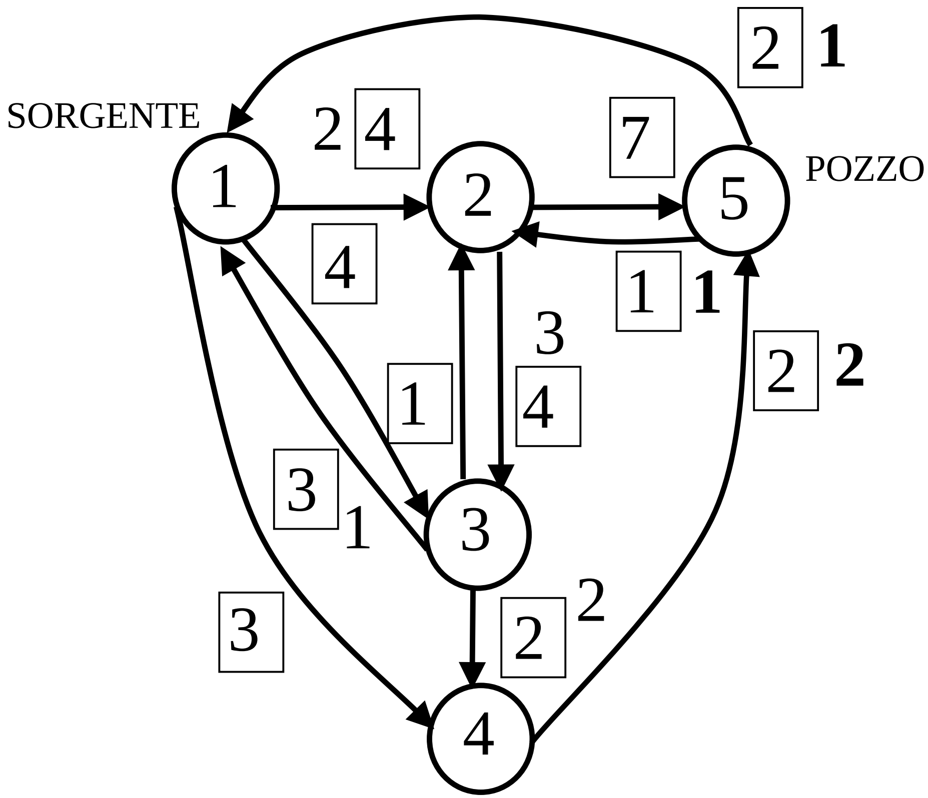
Es. 4: In tabella sono riportati gli archi di una rete di flusso con 5 nodi, in più sono dati i valori di capacità di ogni arco e i flussi di una distribuzione di flusso iniziale. Domande:

1. Partendo dalla soluzione data in tabella, si determini una soluzione ottima al problema di massimo flusso tra i nodi 1 a 5
2. Individuare un taglio di capacità minima tra i nodi 1 e 5
3. Mostrare valore del massimo flusso e capacità del minimo taglio

Archi	(1,2)	(1,3)	(1,4)	(2,3)	(2,5)	(3,1)	(3,2)	(3,4)	(4,5)	(5,1)	(5,2)
Flussi	2	0	0	3	0	1	0	2	2	1	1
Capacità	4	4	3	4	7	3	1	2	2	2	1

Esercizi Algo di Ford–Fulkerson (34)

Archi	(1,2)	(1,3)	(1,4)	(2,3)	(2,5)	(3,1)	(3,2)	(3,4)	(4,5)	(5,1)	(5,2)
Flussi	2	0	0	3	0	1	0	2	2	1	1
Capacità	4	4	3	4	7	3	1	2	2	2	1

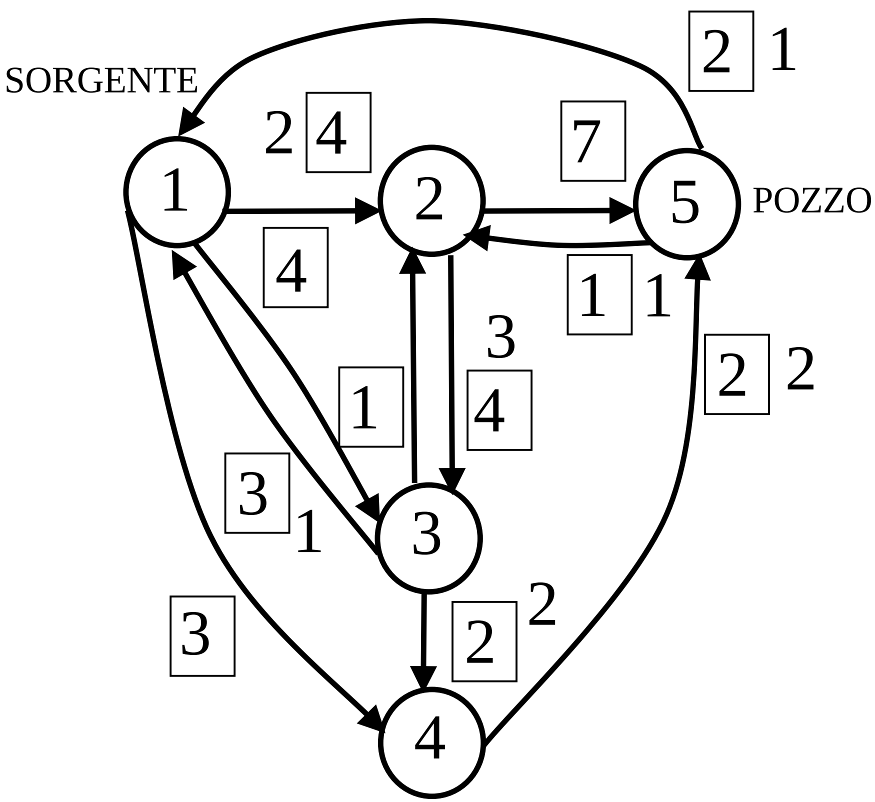


Il <u>flusso iniziale è 0</u> (flusso entrante nel pozzo – flusso uscente dal pozzo)

<u>Nodo 5</u>: entra flusso 2 esce flusso 1+1

I vincoli di bilanciamento delle masse: nodo 2 entra 2+1 esce 3, nodo 3 entra 3 esce 1+2, nodo 4 entra 2 esce 2

Esercizi Algo di Ford–Fulkerson (35)



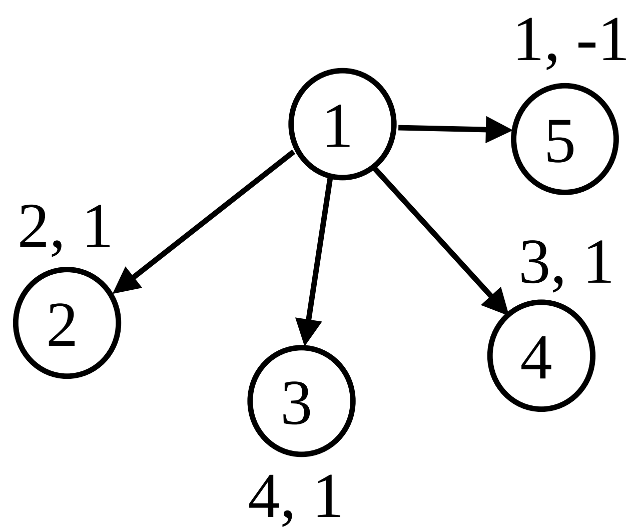
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

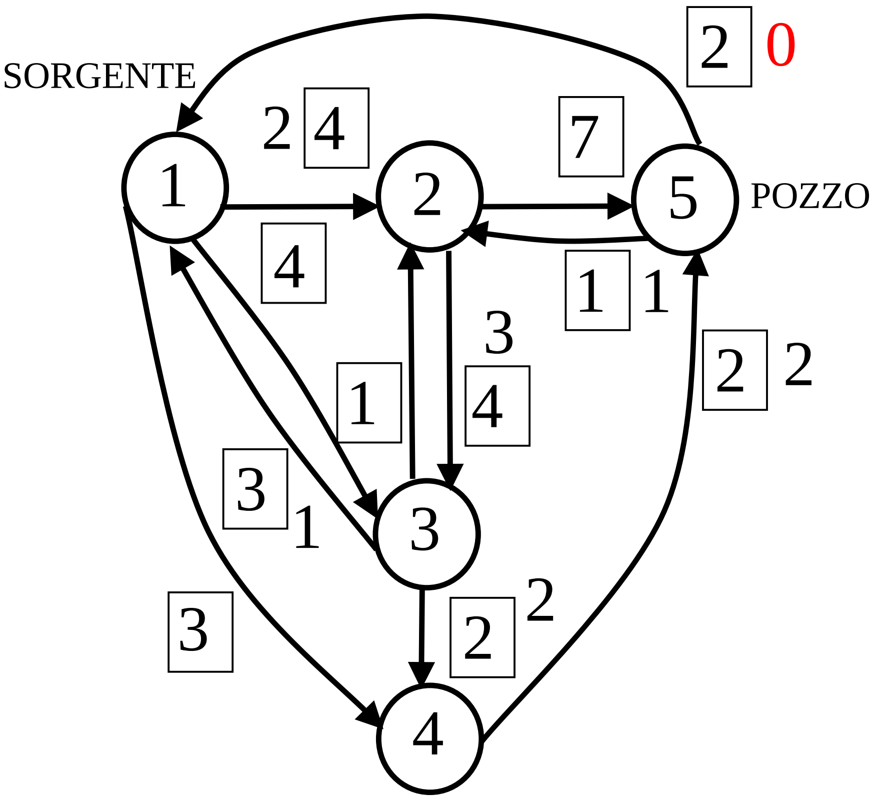
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (36)



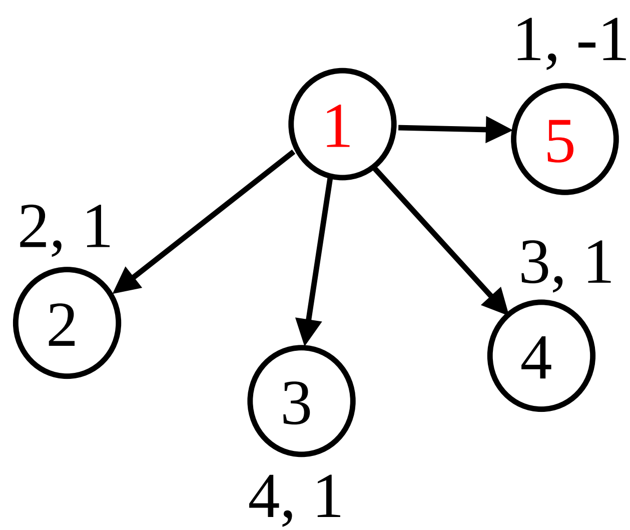
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

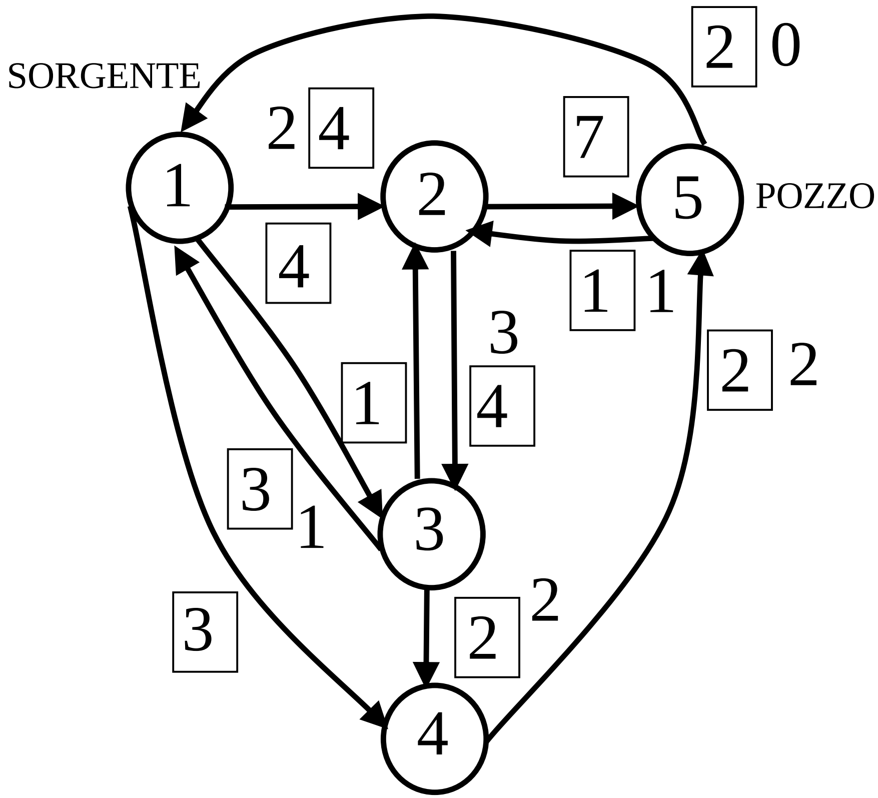
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (37)



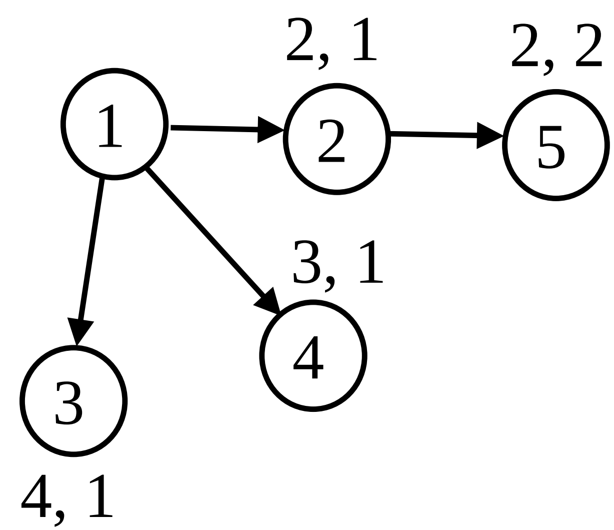
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

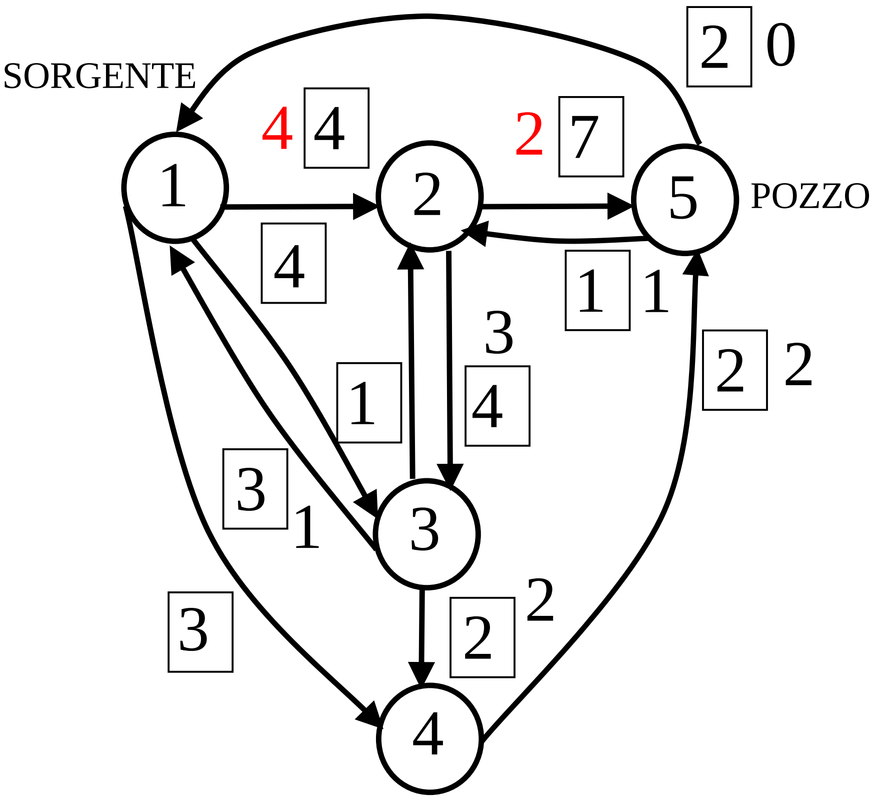
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (38)



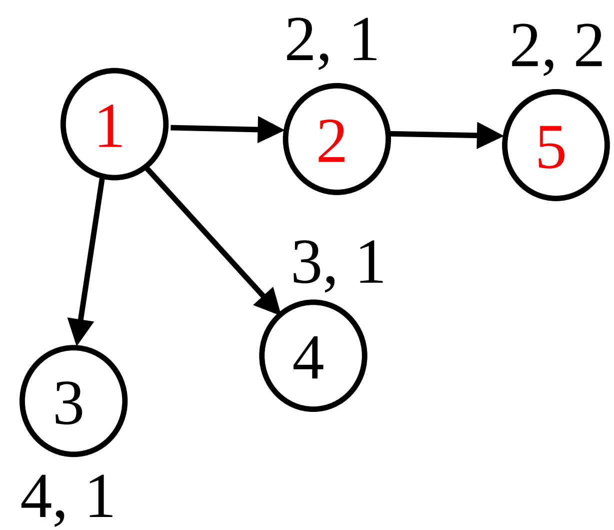
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

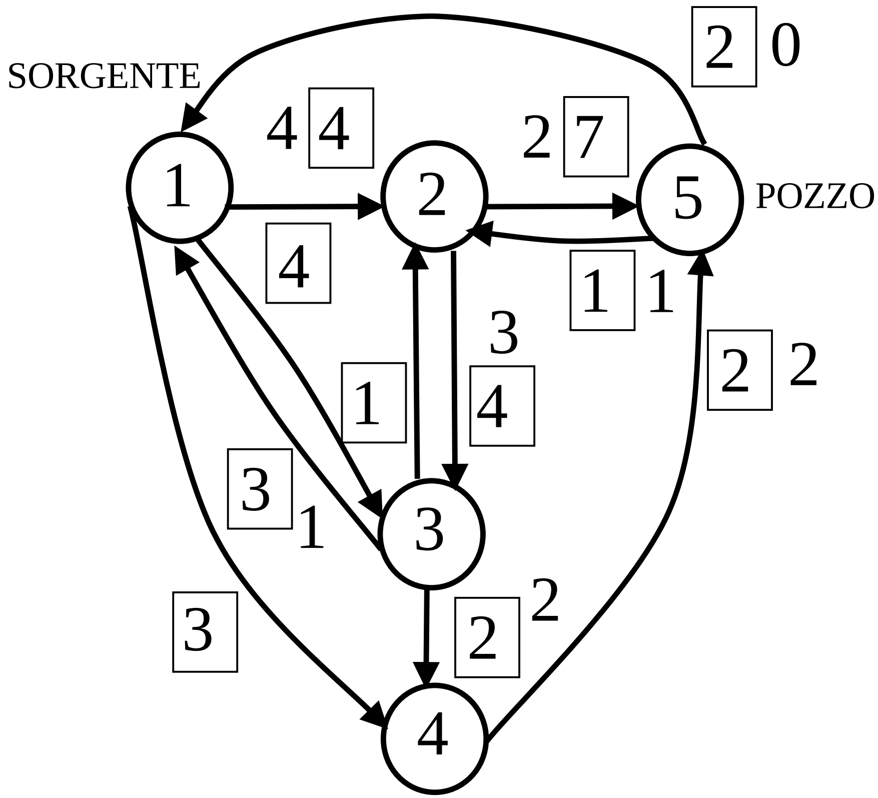
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (39)



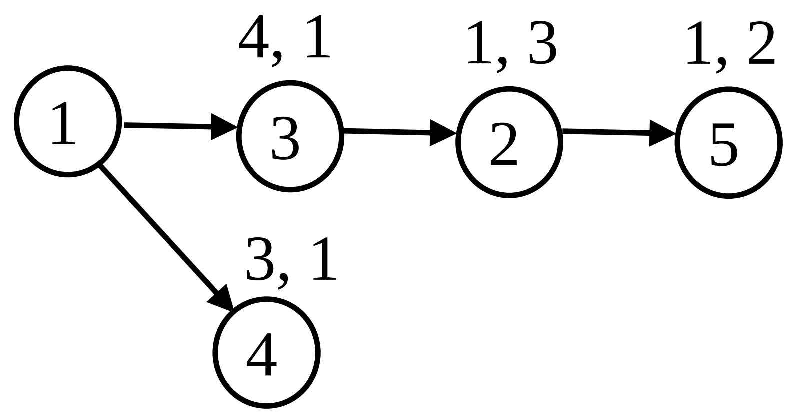
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

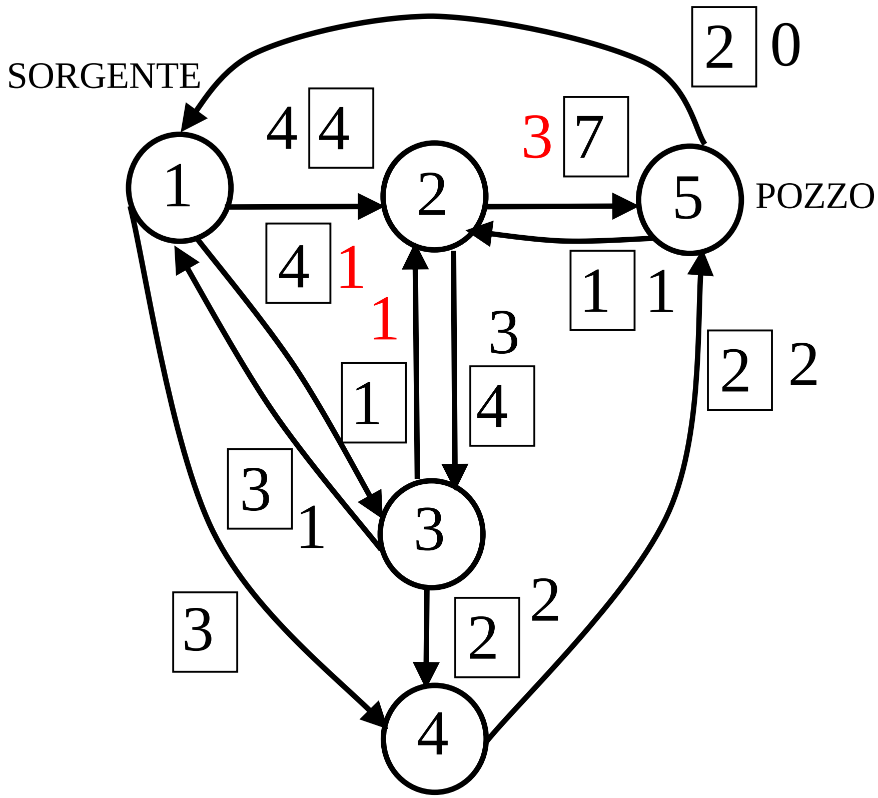
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (40)



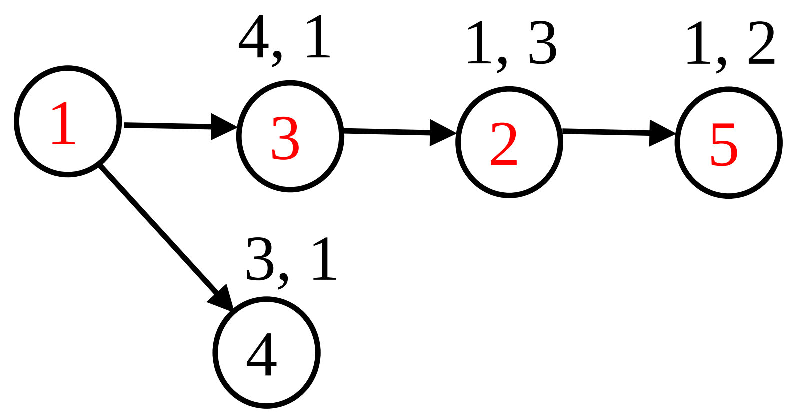
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

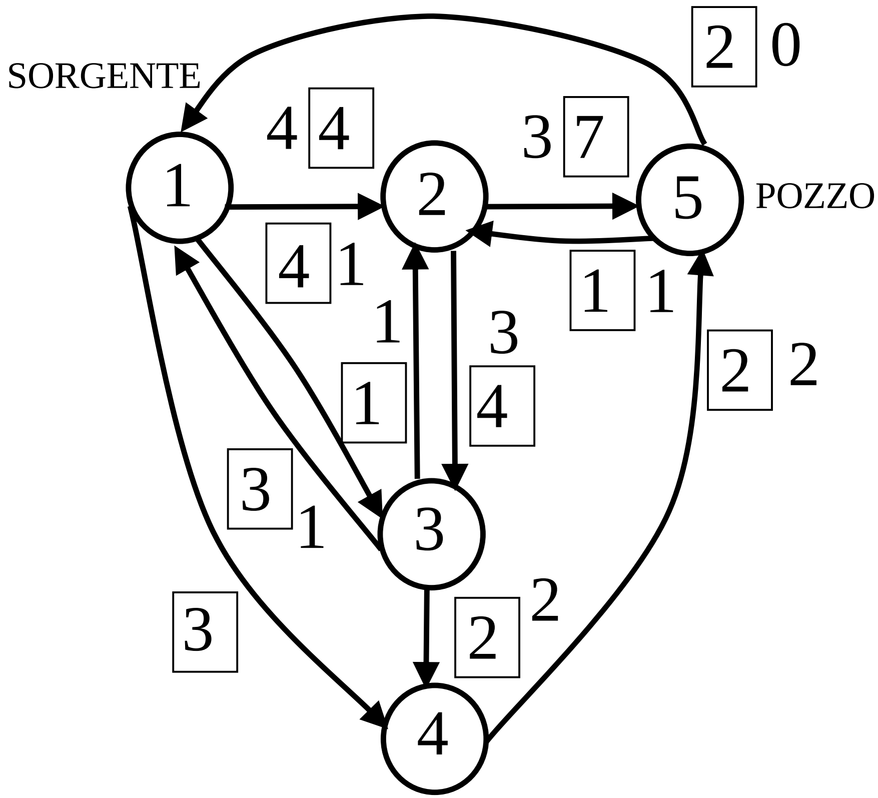
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (41)



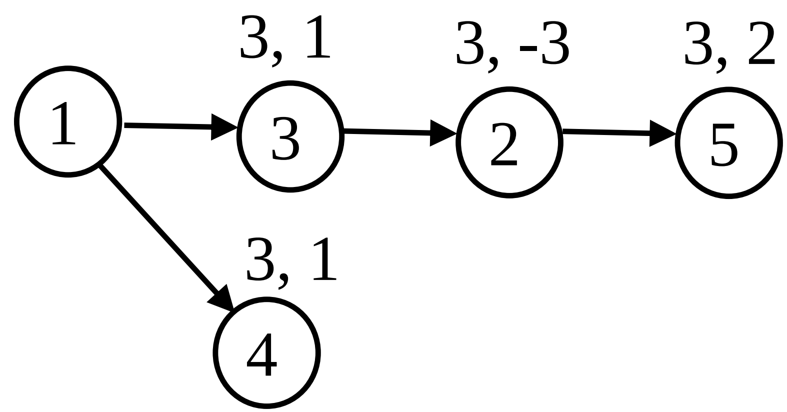
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

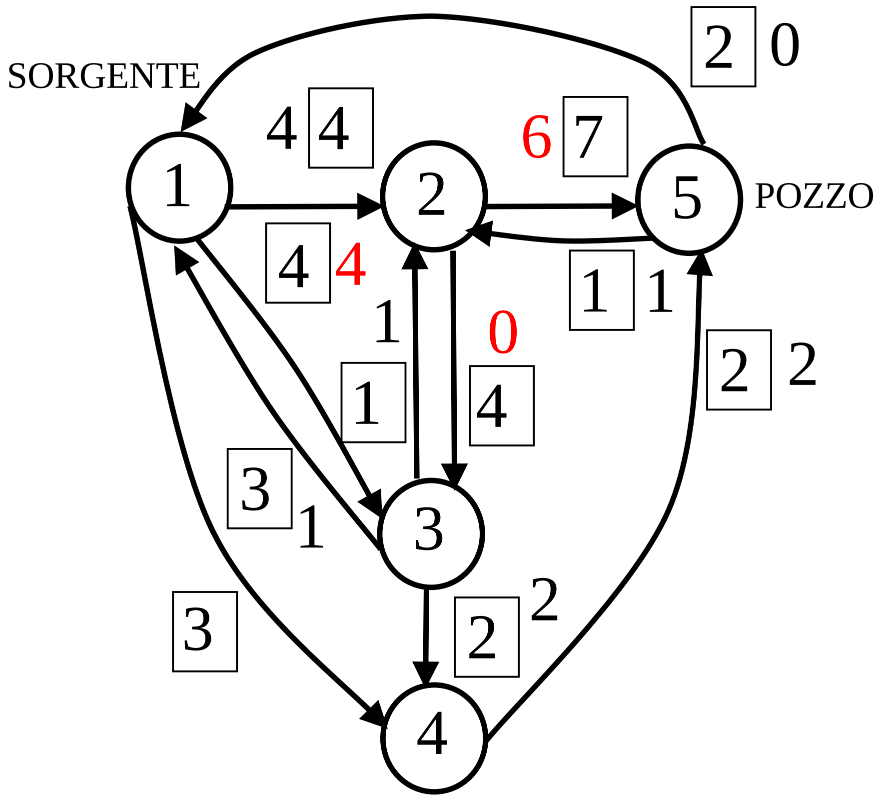
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (42)



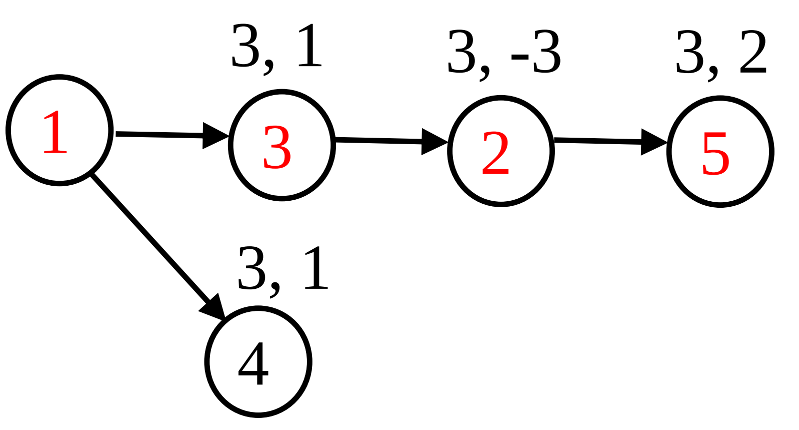
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

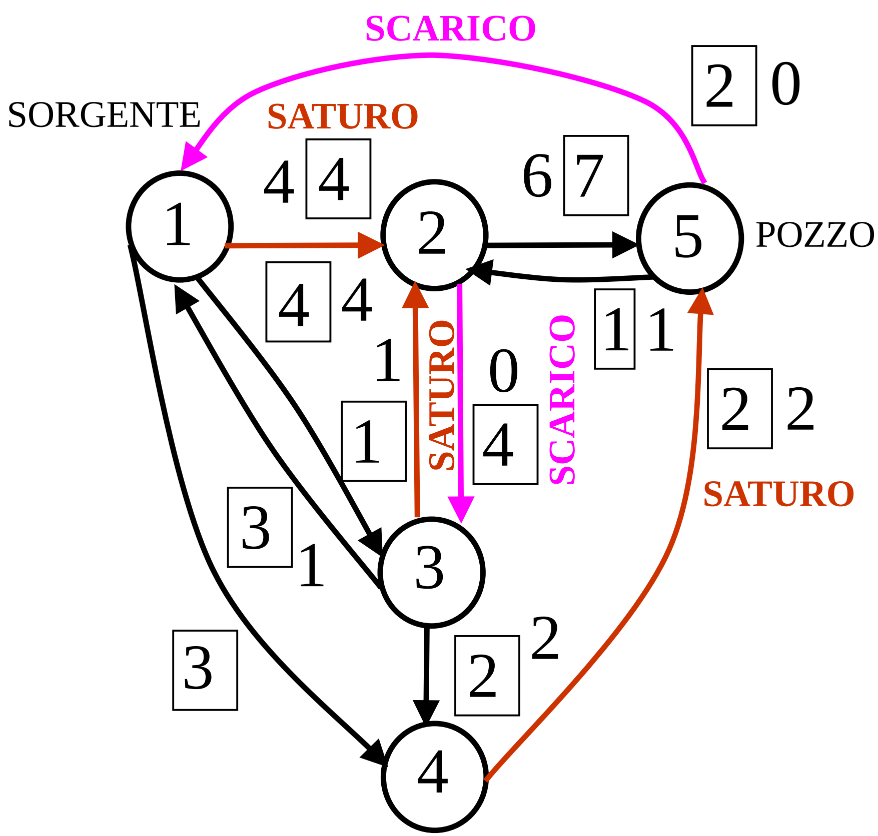
1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3



Esercizi Algo di Ford–Fulkerson (43)



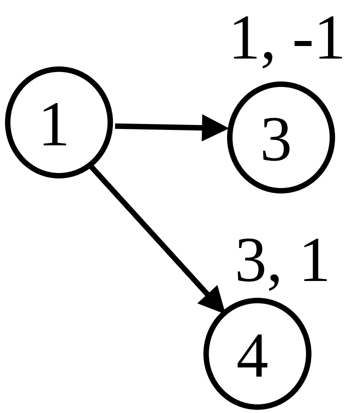
Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

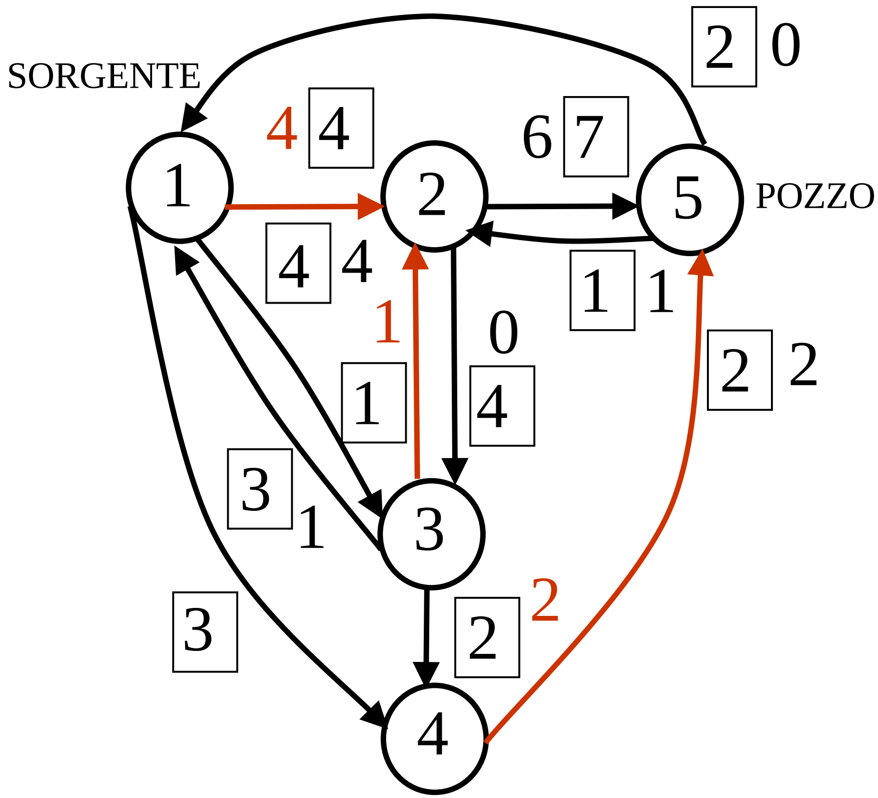
1→3←2→5 con flusso aumentante 3



S **= [1,3,4]**

S̲ **= [2,5]**

Esercizi Algo di Ford–Fulkerson (44)



S = [1,3,4]

$\underline{S}$ = [2,5]

Troviamo i cammini aumentanti

1←5 con flusso aumentante 1

1→2→5 con flusso aumentante 2

1→3→2→5 con flusso aumentante 1

1→3←2→5 con flusso aumentante 3

Totale flusso aumentante = 7

La ricerca di un cammino aumentante si arresta evidenziando il taglio {1,3,4} formato dagli archi (1,2), (3,2),(4,5) di capacità 4+1+2=7

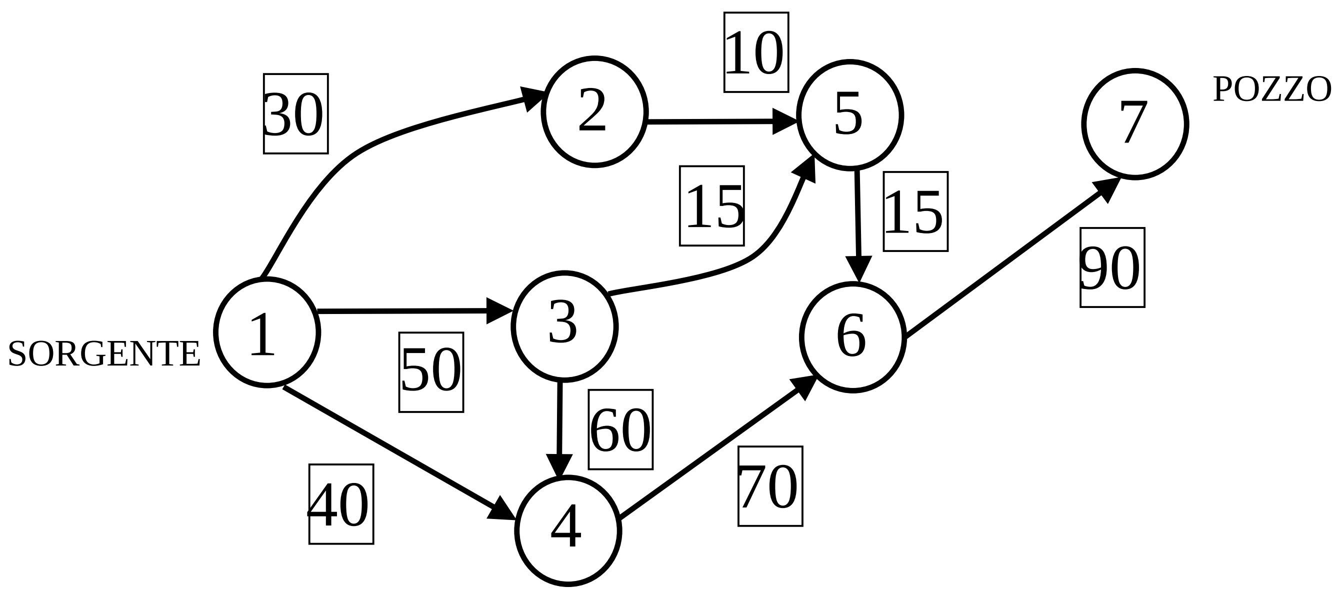
Da cui ottimalità di taglio e flusso

Esercizi Algo di Ford–Fulkerson (45)

Es. 5: In tabella sono riportati gli archi di un digrafo *G* con 7 nodi, e sono dati i costi di ogni arco. Domande:

1. Risolvere il problema del massimo flusso da 1 a 7 con Ford e Fulkerson
2. Mostrate il taglio a capacità minima
3. Dire che cosa accade se la capacità dell'arco (3,5) viene ridotta a zero

Archi	(1,2)	(1,3)	(1,4)	(2,5)	(3,4)	(3,5)	(4,6)	(5,6)	(6,7)
Capacità	30	50	40	10	60	15	70	15	90



Esercizi Algo di Ford–Fulkerson (46)

Partiamo dal digrafo completamente scarico (rete vuota):

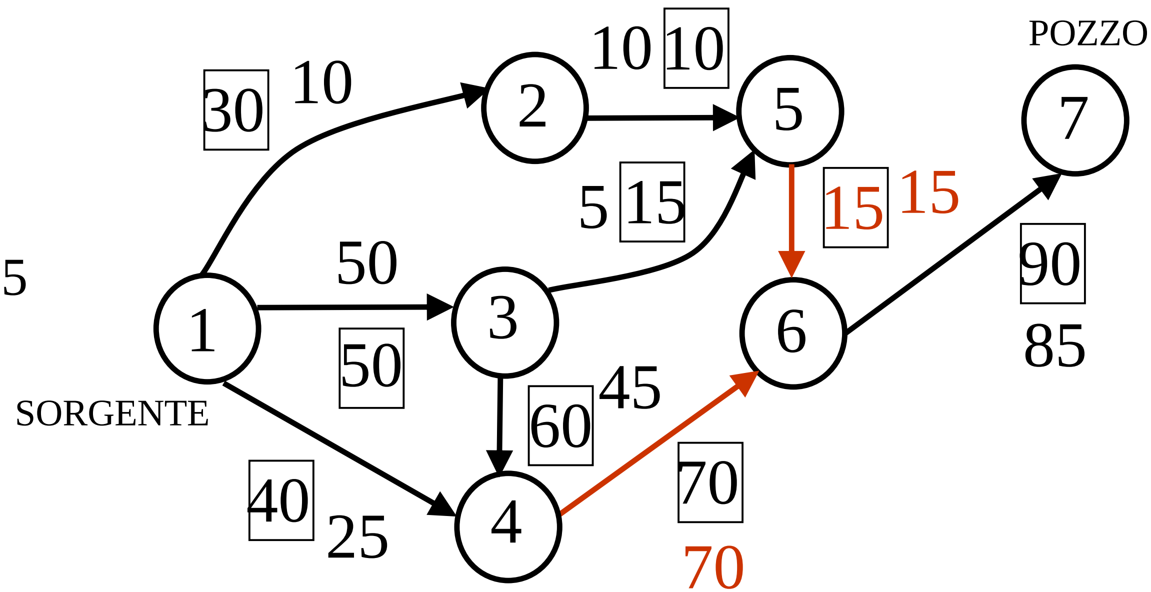
1. I cammini aumentanti sono:

1→2→5→6→7 flusso 10

1→3→4→6→7 flusso 50

1→4→6→7 flusso 20

1→4←3→5→6→7 flusso 5



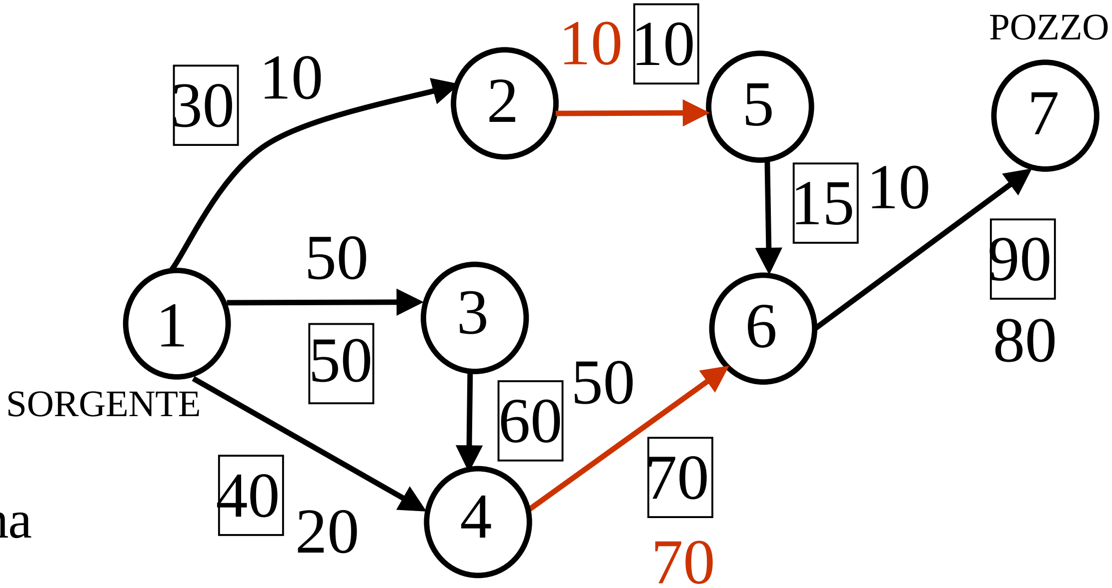
2. Il flusso massimo individuato è pari a 85, ed il taglio di capacità minima è (2,5), (4,6), (3,5)

Esercizi Algo di Ford–Fulkerson (47)

3. Capacità dell'arco (3,5) viene ridotta a zero

⇒ il cammino aumentante
1→4←3→5→6→7
non c'è !

⇒ il taglio di capacità minima diventa (2,5), (4,6) ed il flusso massimo diventa 80



Esercizi su problemi di massimo flusso (48)

Es. 6:

La produzione del pane su scala industriale segue un processo in 5 fasi:

(1) preparazione ingredienti, (2) impasto in gradienti, (3) prima cottura, (4) trattamento superficiale pane, (5) seconda cottura.

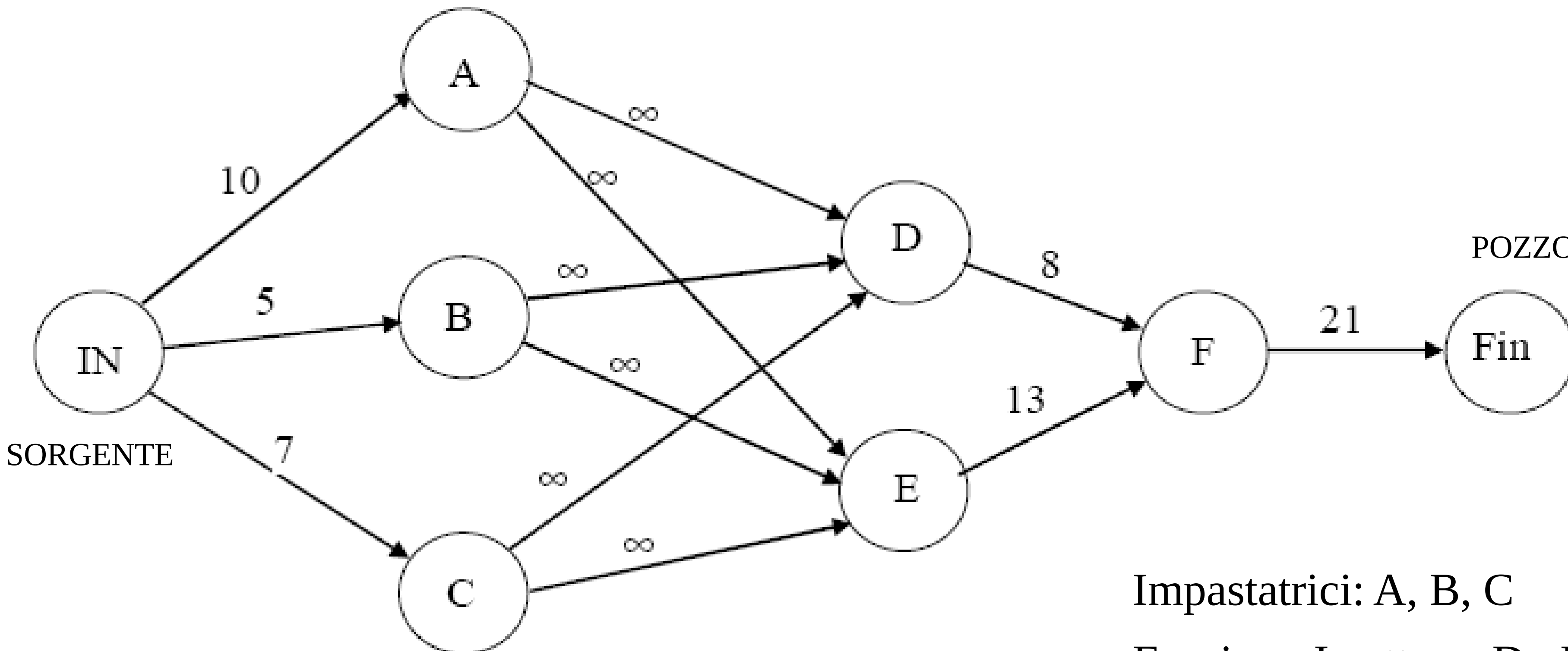
Un forno dispone di:

- 3 impastatrici A,B,C di capacità (in kg di farina per ora) 10, 5, 7 risp.;
- 2 forni D,E per I cottura di capacità (in kg di farina per ora) 8, 13 risp.;
- 1 forno F per II cottura di capacità (in kg di farina per ora) 21;
- Personale largamente sufficiente a seguire tutte le lavorazioni manuali.

<u>Domanda</u>: Si vuole determinare la produzione massima del forno (in kg di farina per ora)

Esercizi Algo di Ford–Fulkerson (49)

Es. 6: Risolvere il problema di massimo flusso su un digrafo:

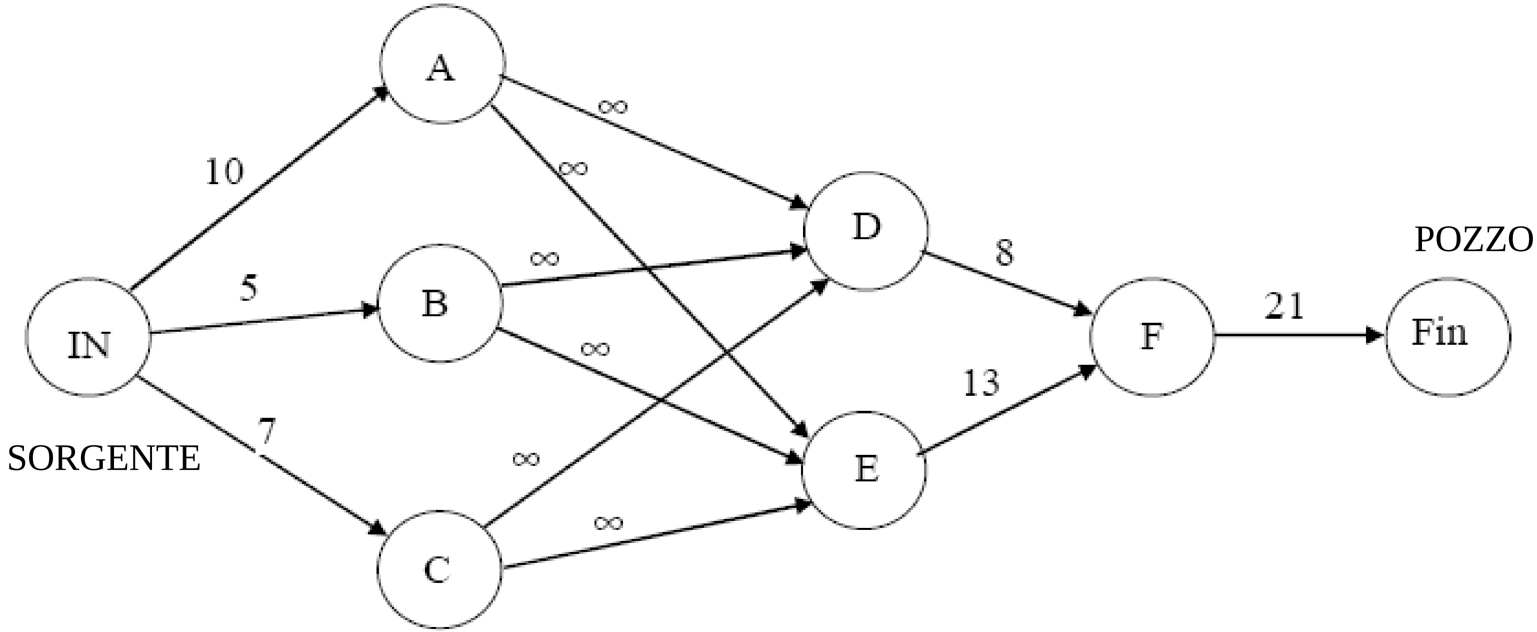


Impastatrici: A, B, C

Forni per I cottura: D , E

Forno per II cottura: F

Esercizi Algo di Ford–Fulkerson (50)



La ricerca di cammini aumentanti (a partire da un flusso iniziale nullo) produce:

cammino (IN) A D F (Fin) flusso aumentante 8

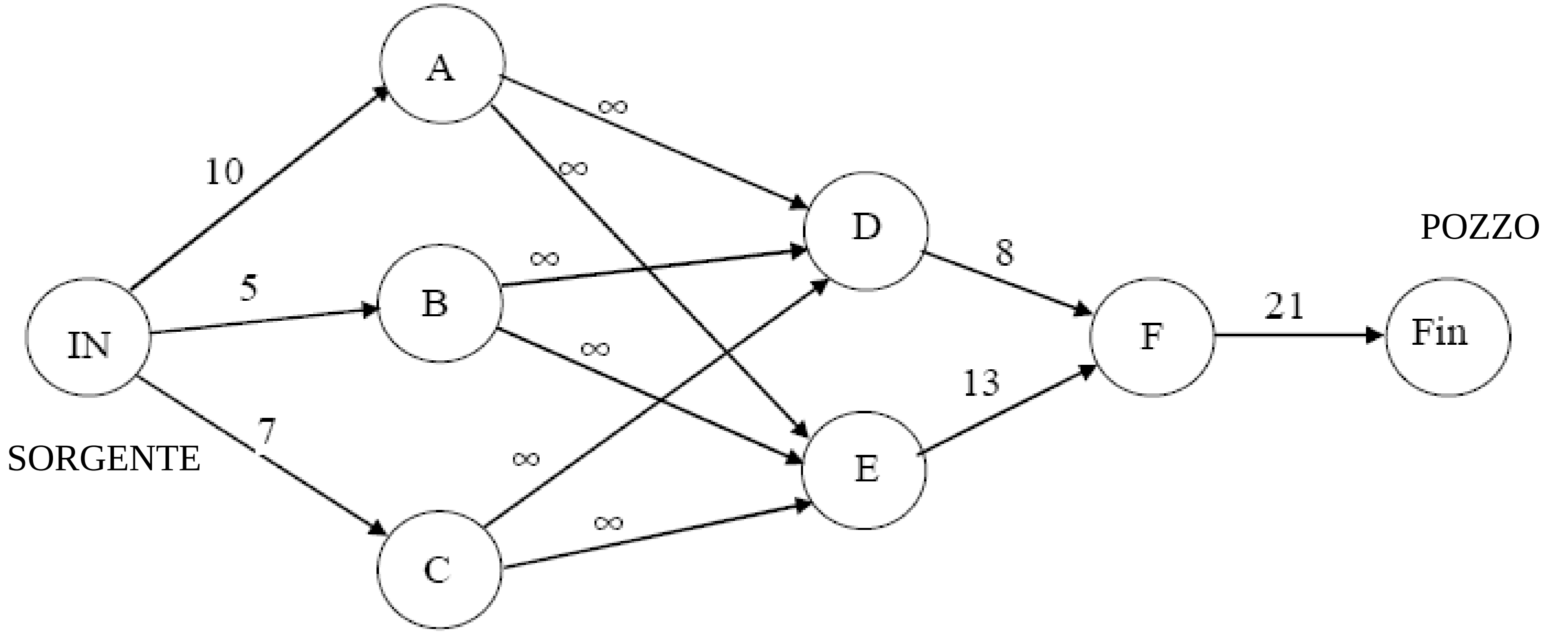
cammino (IN) A E F (Fin) flusso aumentante 2

cammino (IN) B E F (Fin) flusso aumentante 5

cammino (IN) C E F (Fin) flusso aumentante 6

$\Rightarrow$ non ci sono più cammini aumentanti, il flusso corrente 21 è massimo

Esercizi Algo di Ford–Fulkerson (51)



⇒ non ci sono più cammini aumentanti, il flusso corrente 21 è massimo

⇒ per certificare l'ottimalità della soluzione basta osservare che il taglio S = {IN, A, B, C, D, E }, <u>S</u> ={F, Fin} ha capacità 21

(ed è quindi il taglio minimo: flusso massimo = taglio corrente)