Lezioni di Ricerca Operativa

Corso di Laurea in Informatica

Università di Salerno

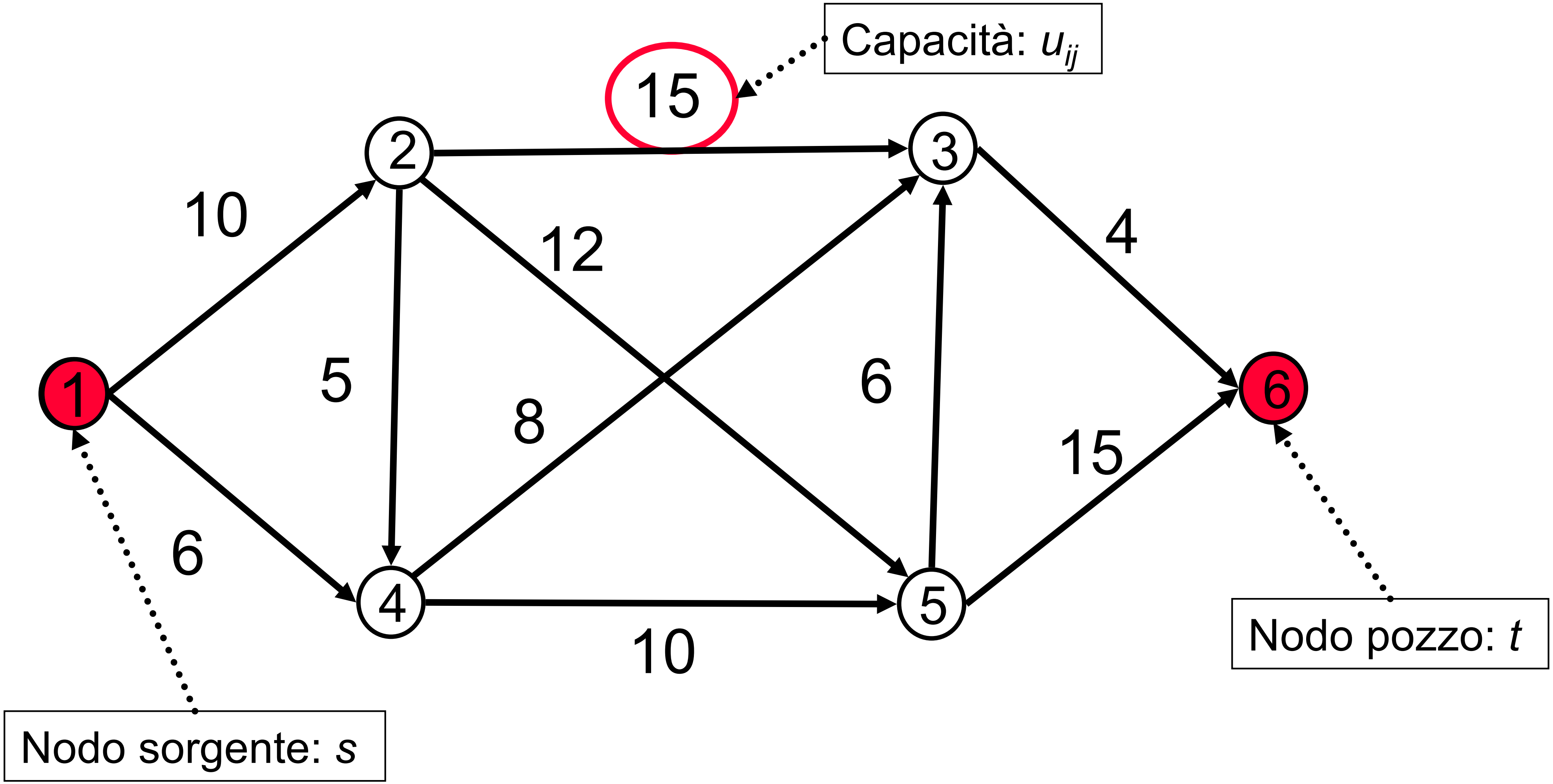
Problema del Massimo Flusso:

- Formulazione Matematica
- Teorema flusso massimo / taglio minimo
- Algoritmo del Grafo Ausiliario

R. Cerulli – F. Carrabs

Il Problema del Massimo Flusso

Sia G = (V,A) un grafo orientato su cui sia definito un vettore $\underline{u} = [u_{ij}]$ delle capacità associate agli archi del grafo; inoltre, siano *s* e *t* due nodi distinti, detti rispettivamente *sorgente (o origine)* e *pozzo (o destinazione)*. Il problema del flusso massimo consiste nel determinare la massima quantità di flusso che è possibile inviare da s a t attraverso G.



Il Problema del Massimo Flusso

Nodo sorgente fornisce flusso → f
Nodo destinazione assorbe flusso→ $-f$
Tutti gli altri nodi sono nodi di transito

Voglio spedire dalla sorgente la massima quantità di flusso f fino al pozzo senza violare i vincoli di capacità

Il Problema del Massimo Flusso: formulazione

Vincoli di **bilanciamento del flusso**

$$\max f$$

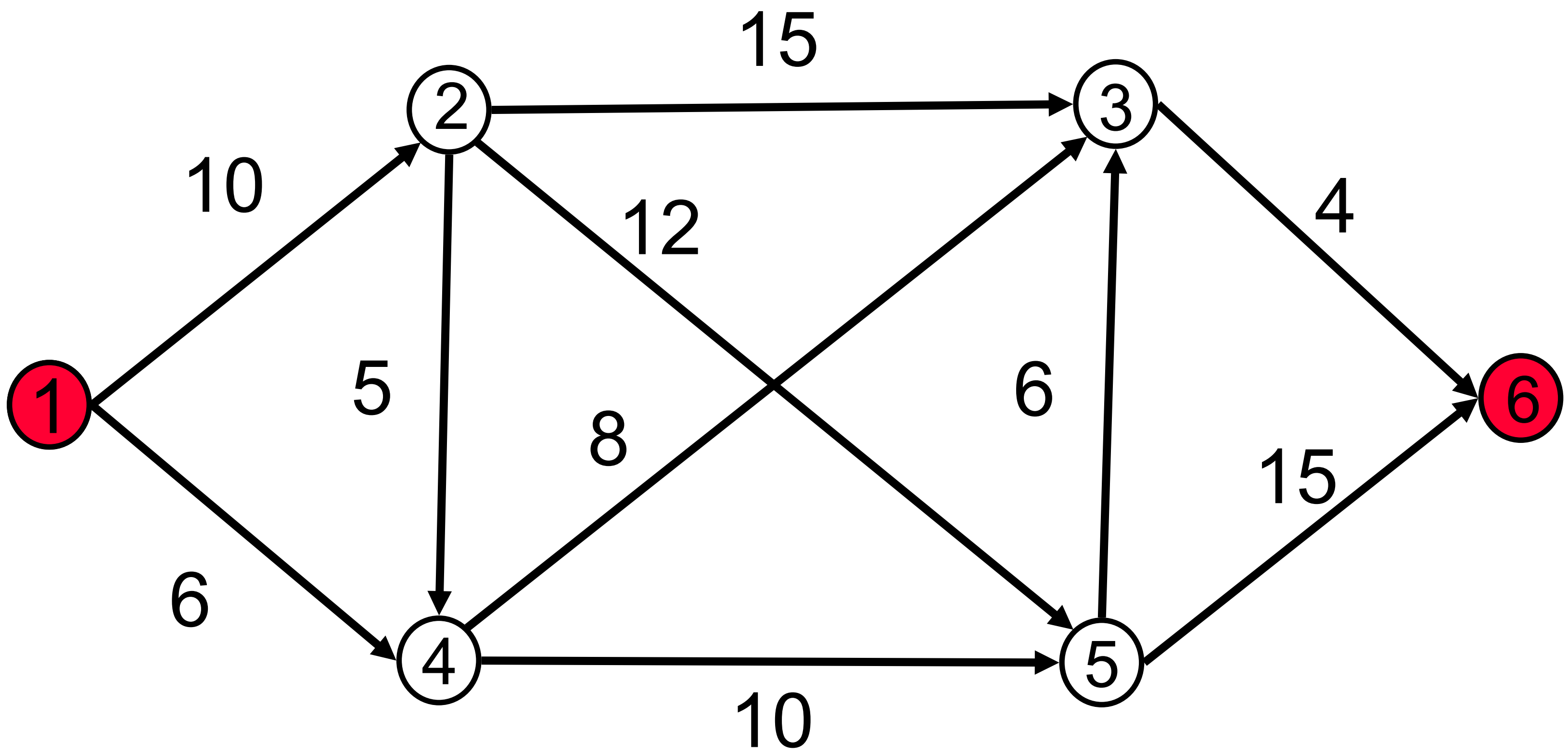
con vincoli :

$$\sum_{j \in FS(i)} x_{ij} - \sum_{k \in BS(i)} x_{ki} = \begin{cases} 0 & \forall i \in V \; i \neq s,t \\ f & se \quad i = s \\ -f & se \quad i = t \end{cases} \qquad (1)$$

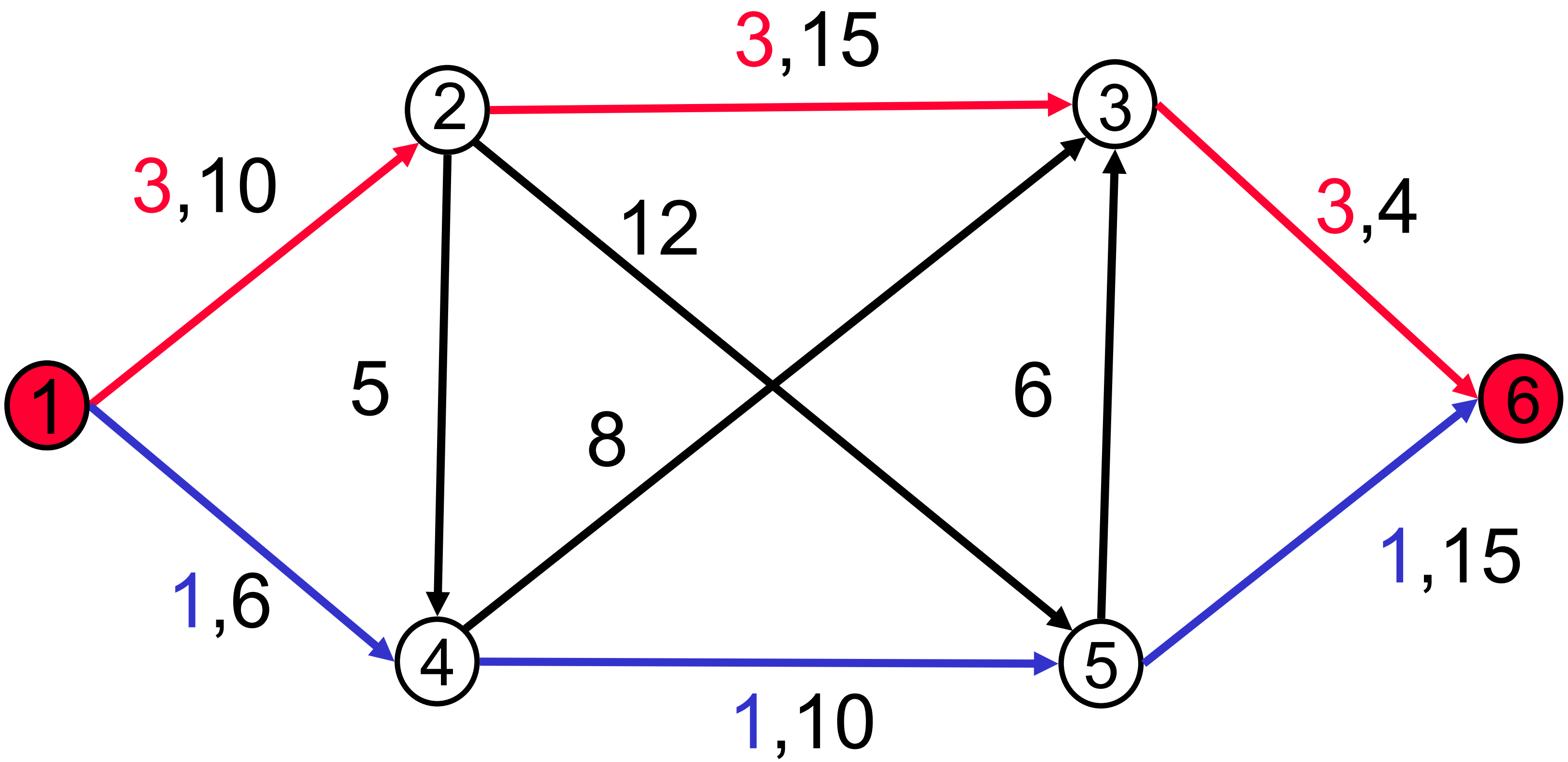
$$0 \leq x_{ij} \leq u_{ij} \qquad \forall \, (i,j) \in A \qquad (2)$$

Vincoli di **capacità**

Il Problema del Massimo Flusso: esempio

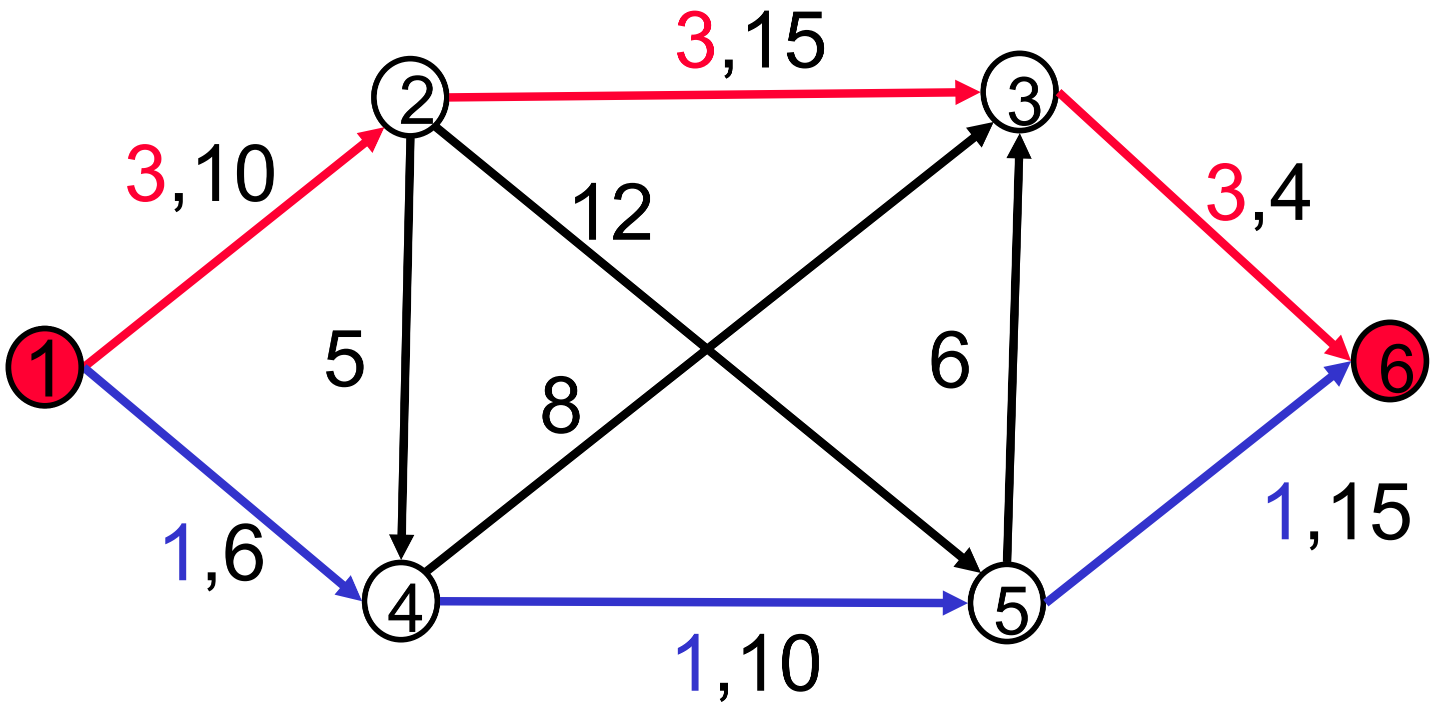


Il Problema del Massimo Flusso: esempio



$$x_{12} = 3 \quad x_{23} = 3 \quad x_{36} = 3 \quad x_{14} = 1 \quad x_{45} = 1 \quad x_{56} = 1$$

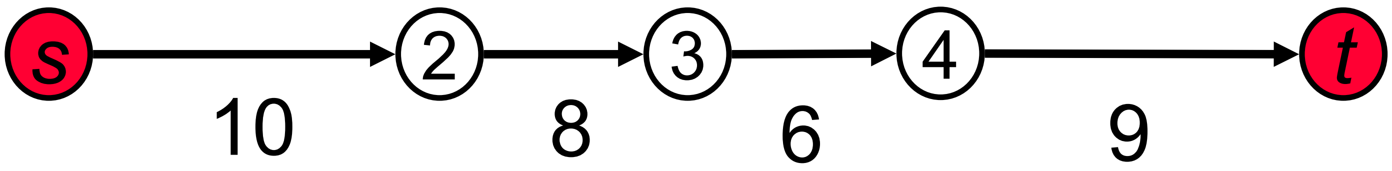
Il Problema del Massimo Flusso: esempio



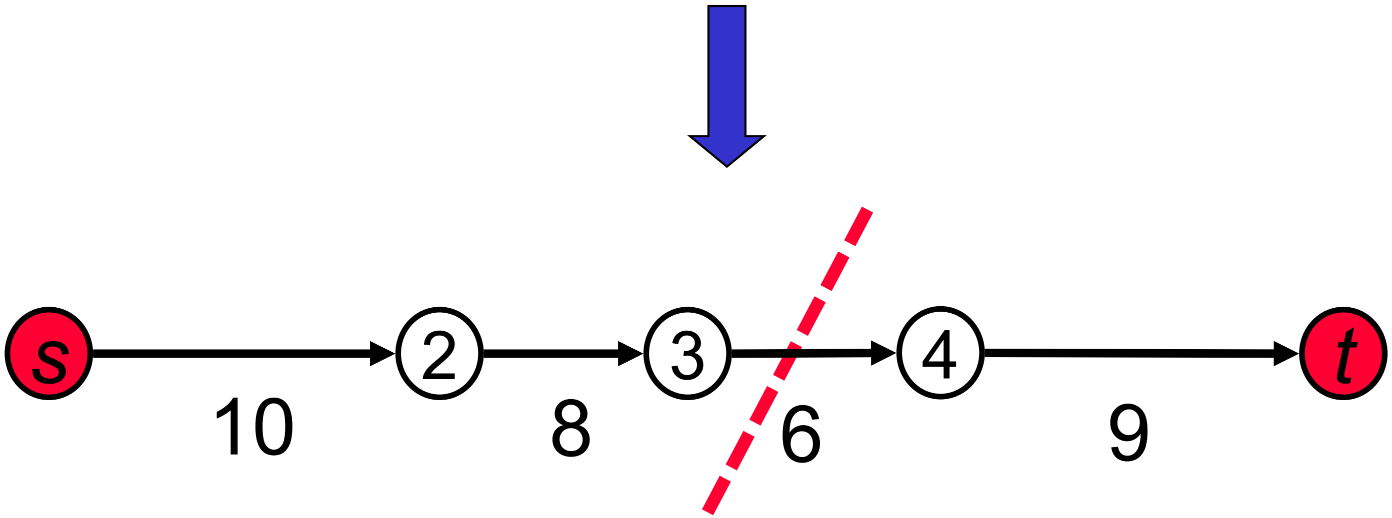
$$\underline{x} = \begin{bmatrix} x_{12} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{36} \\ x_{43} \\ x_{45} \\ x_{53} \\ x_{56} \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 3 \\ 0 \\ 0 \\ 3 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$\underline{x}$ rappresenta un **flusso ammissibile,** con valore f=4, per il grafo G.

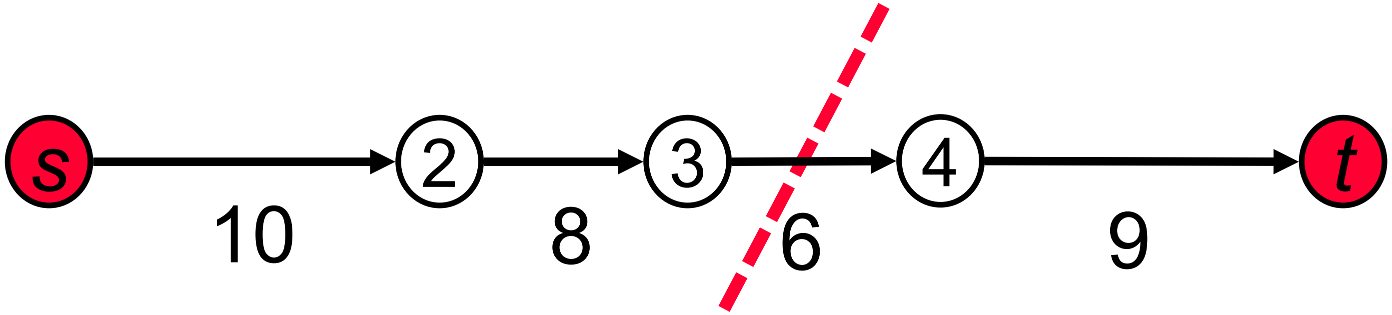
Il Problema del Massimo Flusso: concetti fondamentali



Il flusso massimo su questo grafo è pari a 6 (corrispondente alla capacità minima degli archi del cammino).



Il Problema del Massimo Flusso: concetti fondamentali



Il segmento tratteggiato in figura mostra un **taglio s-t** del grafo, ossia un **partizionamento dei vertici** del grafo in due sottoinsiemi V_1={s,2,3} e V_2={4,t} tali che:

- Il nodo sorgente appartiene a V_1
- Il nodo pozzo appartiene a V_2
- $V_1 \cup V_2 = V$
- $V_1 \cap V_2 = \emptyset$

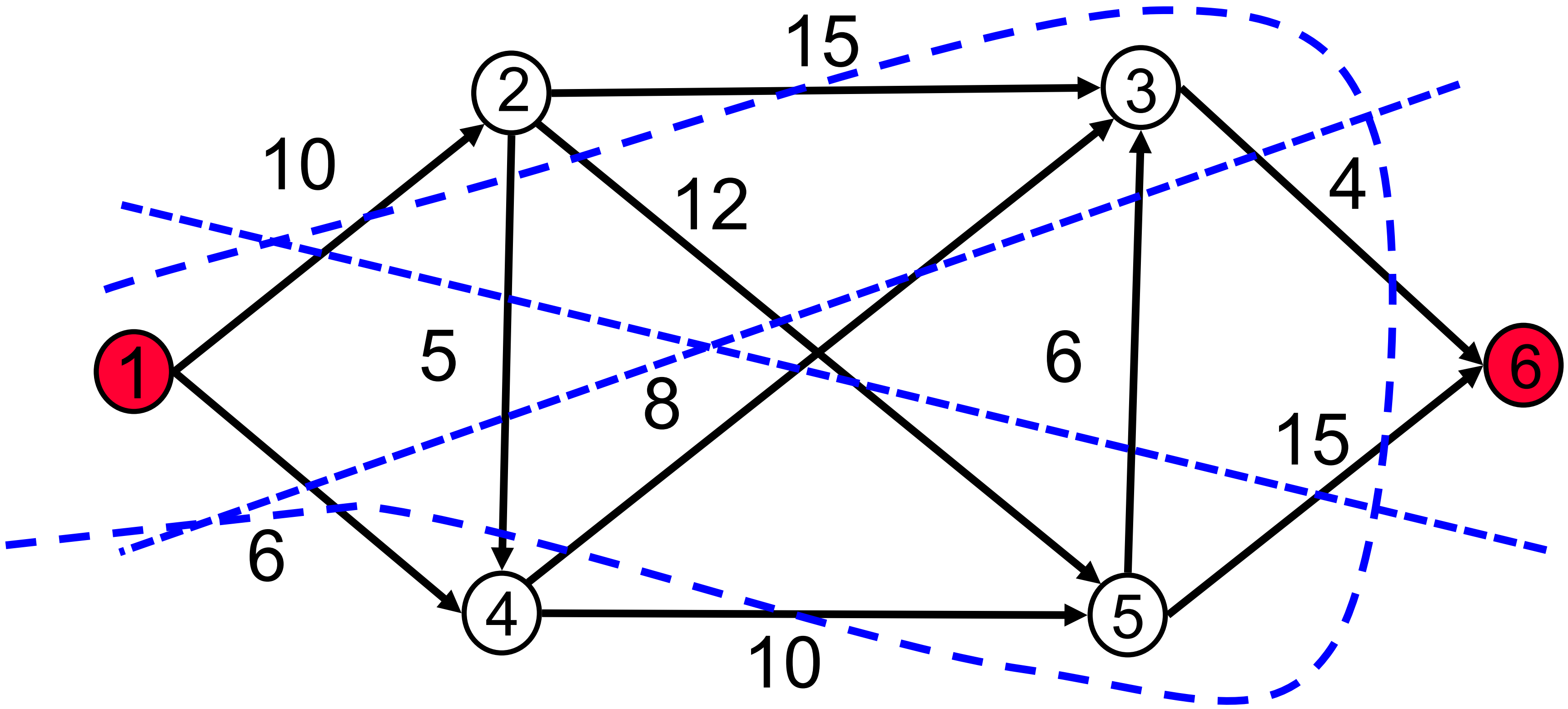
Definizione:

- **Archi diretti** del taglio $[V_1,V_2]$: $\{(i,j): i \in V_1 \text{ e } j \in V_2\}$
- **Archi inversi** del taglio $[V_1,V_2]$: $\{(p,q): p \in V_2 \text{ e } q \in V_1\}$

Estendiamo questo concetto di taglio s-t ad un grafo più complesso.

(nel resto della lezione ometteremo la dicitura "s-t" tenendo presente però che faremo riferimento sempre a tagli di questa tipologia)

Taglio di un grafo e archi del taglio

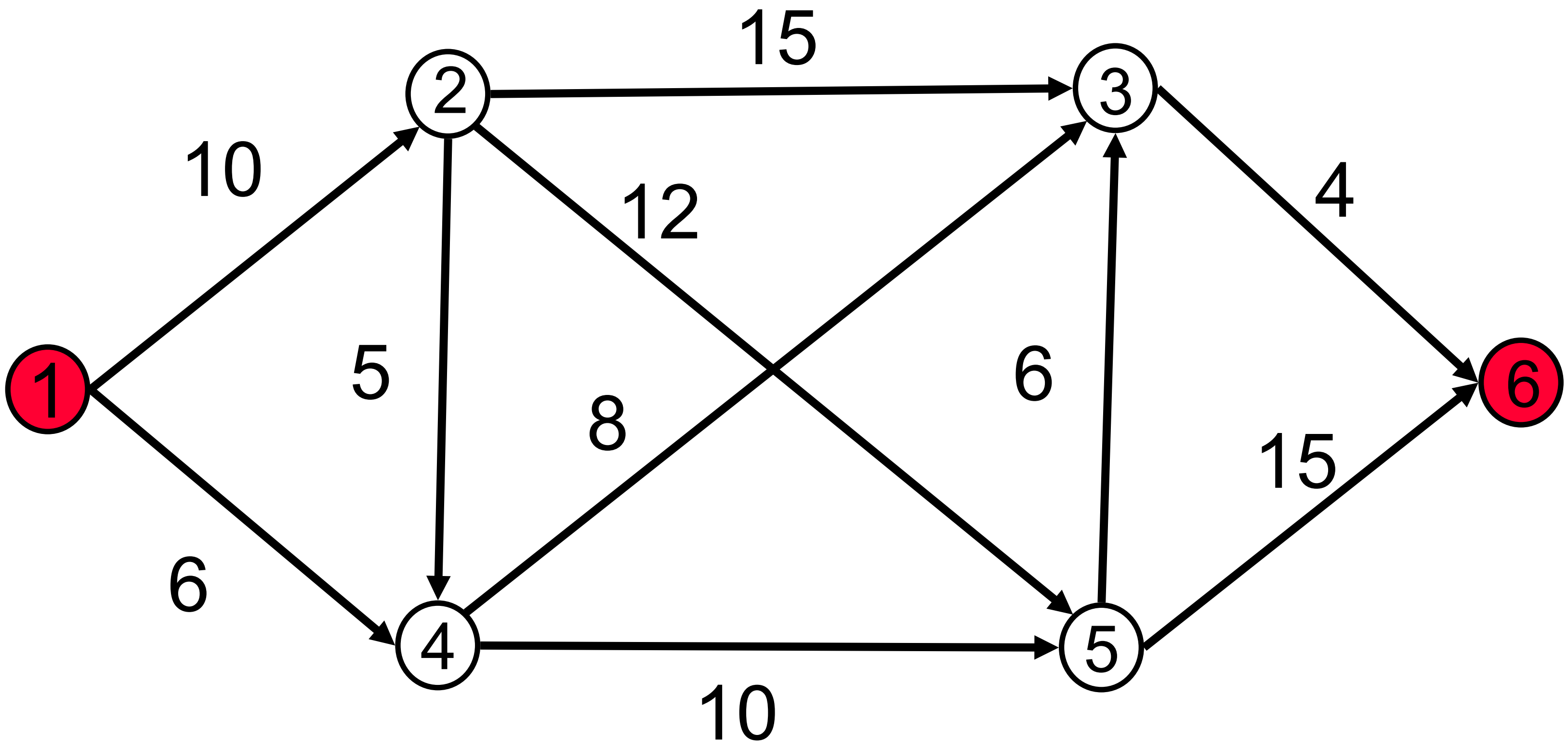


Taglio 1: V_1 ={1,2,3} V_2 = {4,5,6} → archi "diretti" del taglio = {(1,4) (2,4) (2,5) (3,6)}

Taglio 2: V_1 ={1,3,5} V_2 = {2,4,6} → archi "diretti" del taglio = {(1,2) (1,4) (3,6) (5,6)}

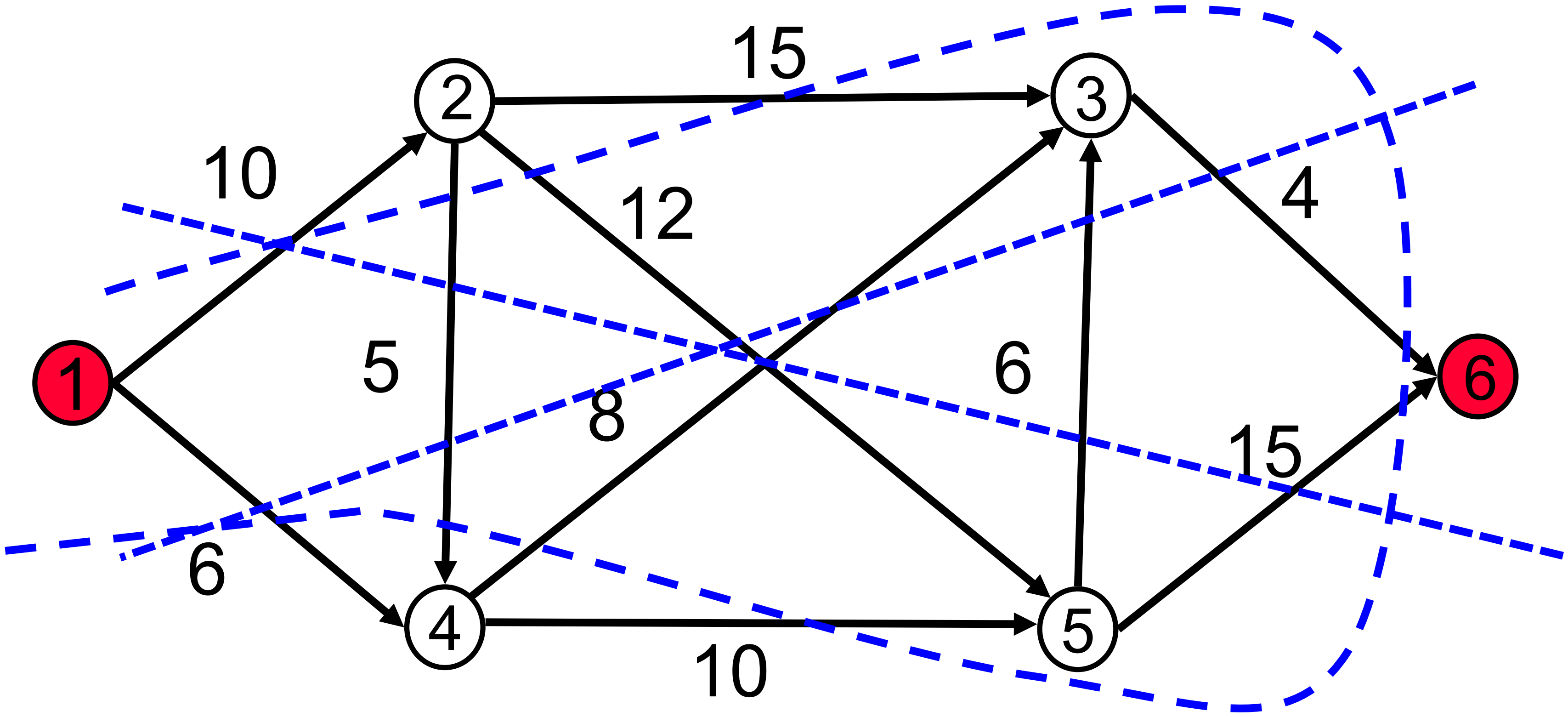
Taglio 3: V_1 ={1,4,5} V_2 = {2,3,6} → archi "diretti" del taglio = {(1,2) (4,3) (5,3) (5,6)}

Capacità di un taglio



Dato il taglio $[V_1,V_2]$ la **capacità del taglio** $u[V_1,V_2]$ è pari alla somma delle capacità degli archi diretti del taglio.

Capacità di un taglio



Taglio 1: V_1 ={1,2,3} V_2 = {4,5,6} → archi diretti del taglio ={(1,4) (2,4) (2,5) (3,6)}
Capacità $u[V_1,V_2]$ = 6 + 5 + 12 + 4 =27

Taglio 2: V_1 ={1,3,5} V_2 = {2,4,6} → archi diretti del taglio ={(1,2) (1,4) (3,6) (5,6)}
Capacità $u\,[V_1,V_2]$ = 10 + 6 + 4 + 15 = 35

Taglio 3: V_1 ={1,4,5} V_2 = {2,3,6} → archi diretti del taglio ={(1,2) (4,3) (5,3) (5,6)}
Capacità $u\,[V_1,V_2]$ = 10 + 8 + 6 + 15 = 39

Relazione tra il massimo flusso e la capacità di un taglio

Proprietà 1:
Il valore di un qualunque flusso ammissibile è minore o uguale alla capacità di un qualunque taglio.

Dim.

Sia $\underline{x}$ un flusso ammissibile e $[V_1,V_2]$ un qualunque taglio del grafo. Sommando i vincoli di bilanciamento del flusso relativi ai nodi in V_1 otteniamo:

$$f = \sum_{i \in V_1} \left[\sum_{j \in FS(i)} x_{ij} - \sum_{k \in BS(i)} x_{ki} \right]$$

Relazione tra il massimo flusso e la capacità di un taglio

Si ha che

$$f = \underbrace{\sum_{i \in V_1} \left(\sum_{j \in FS(i)} x_{ij} - \sum_{k \in BS(i)} x_{ki} \right)}_{(1)} = \sum_{(i,j) \in [V_1, V_2]} x_{ij} - \sum_{(p,q) \in [V_2, V_1]} x_{pq}$$

Infatti:

- Per ogni arco (i,j) con *i* e *j* in V_1, x_{ij} appare due volte in (1), una volta con coefficiente 1 ed una volta con coefficiente -1;
- Per ogni arco (i,j) con *i* in V_1 e *j* in V_2, x_{ij} appare in (1) con coefficiente 1;
- Per ogni arco (i,j) con *i* in V_2 e *j* in V_1, x_{ij} appare in (1) con coefficiente -1;
- Per ogni arco (i,j) con *i* e *j* in V_2, x_{ij} non appare in (1).

Relazione tra il massimo flusso e la capacità di un taglio

Si ha che

$$f = \sum_{i \in V_1} \left(\sum_{j \in FS(i)} x_{ij} - \sum_{k \in BS(i)} x_{ki} \right) = \sum_{(i,j) \in [V_1, V_2]} x_{ij} - \sum_{(p,\, q) \in [V_2, V_1]} x_{pq}$$

$$\Downarrow$$

$$f = \sum_{(i,j) \in [V_1, V_2]} x_{ij} - \sum_{(p,\, q) \in [V_2, V_1]} x_{pq} \leq \sum_{(i,j) \in [V_1, V_2]} x_{ij} \leq \sum_{(i,j) \in [V_1, V_2]} u_{ij} = u[V_1, V_2]$$

Relazione tra il massimo flusso e la capacità di un taglio

La capacità di un taglio fornisce un limite superiore al valore del flusso *f* che posso spedire dalla sorgente al pozzo

Se ho un flusso ammissibile di valore *f* e riesco a trovare un taglio la cui capacità è uguale ad *f* allora posso concludere che il flusso che ho trovato è massimo.

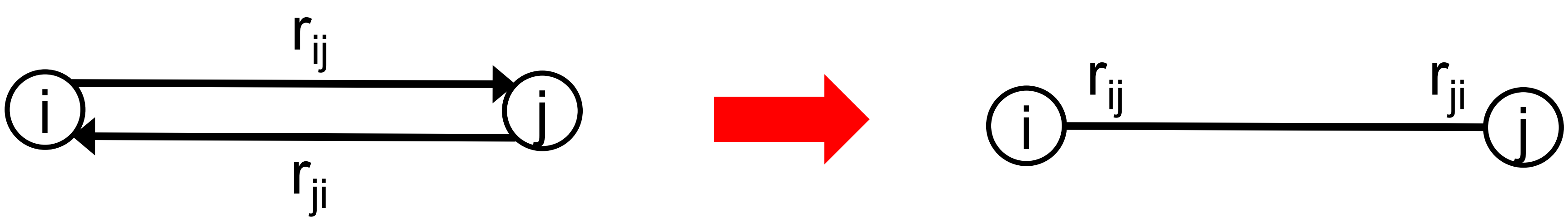
Teorema (Max Flow- Min Cut)
Il flusso massimo che può essere spedito dalla sorgente al pozzo su un grafo orientato G è uguale alla capacità del taglio minimo di G.

Grafo ausiliario e capacità residue

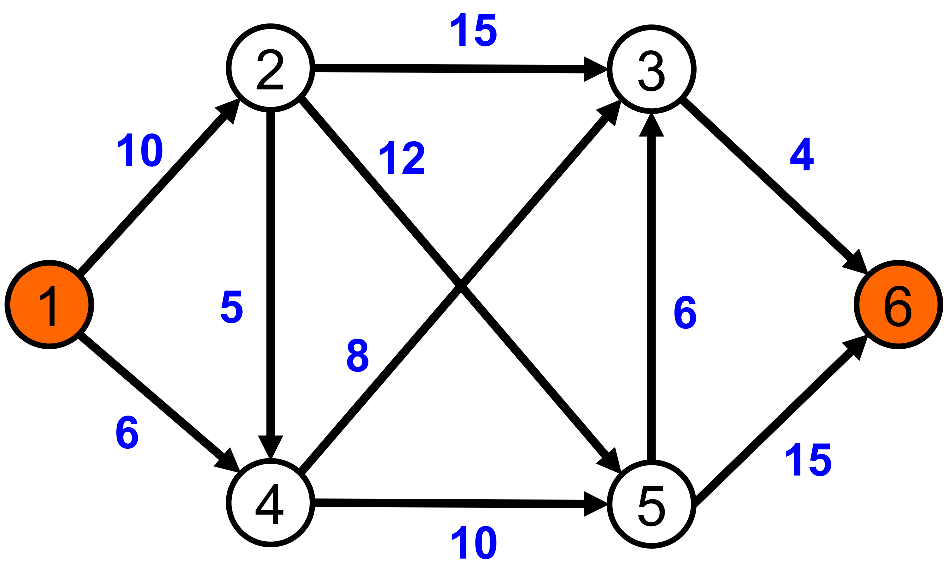
Dato un grafo G=(V,A) ed un flusso ammissibile $\underline{x}$ su G, il **grafo ausiliario** G($\underline{x}$)=(V',A') è così costruito:

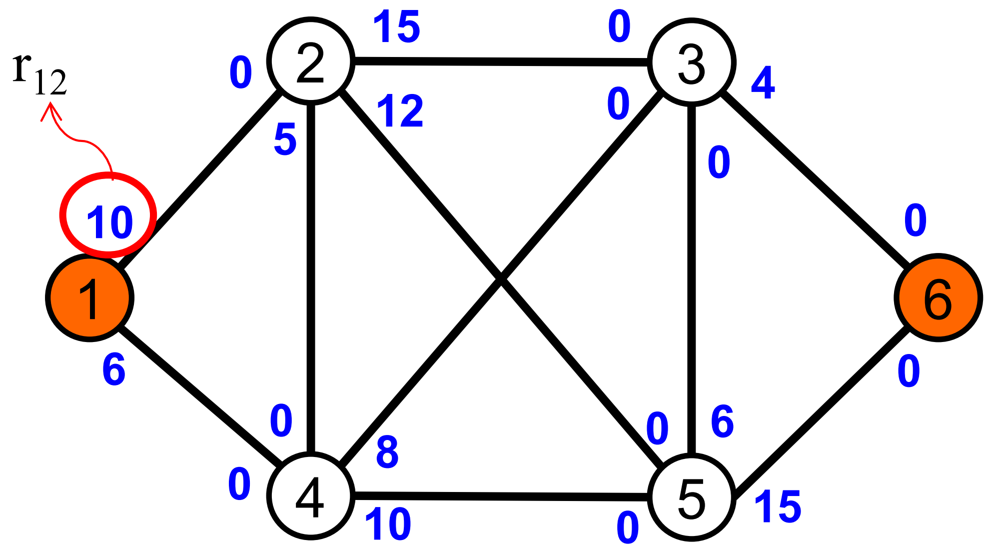
- ✓ V'=V
- ✓ Per ogni arco (i,j) in A, A' contiene gli archi (i,j) e (j,i); la capacità u_{ij} di ogni arco $(i,j) \notin A$ è pari a 0.
- ✓ Ad ogni arco di A' è associata una **capacità residua:** $r_{ij} = u_{ij} - x_{ij} + x_{ji}$.

Utilizzeremo la seguente notazione:



Grafo ausiliario e capacità residue

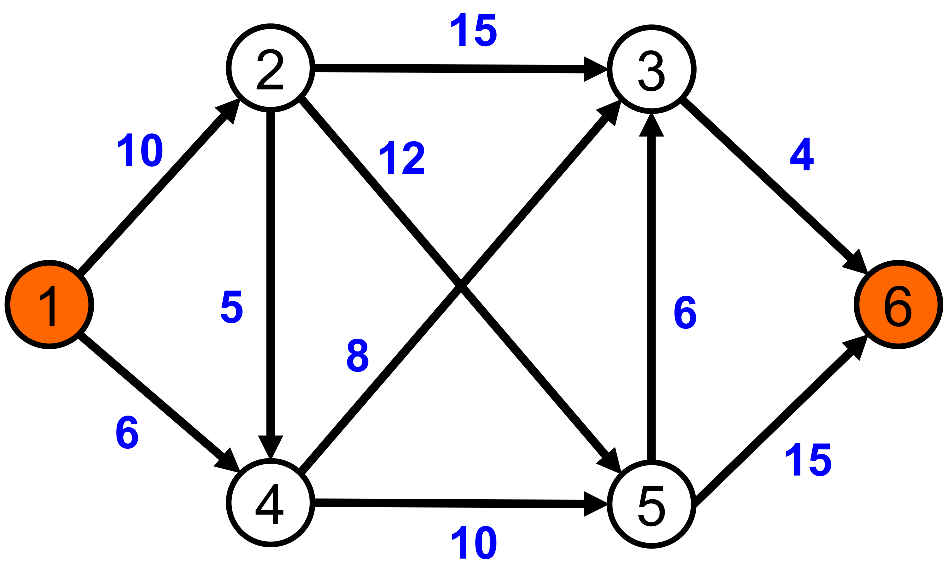


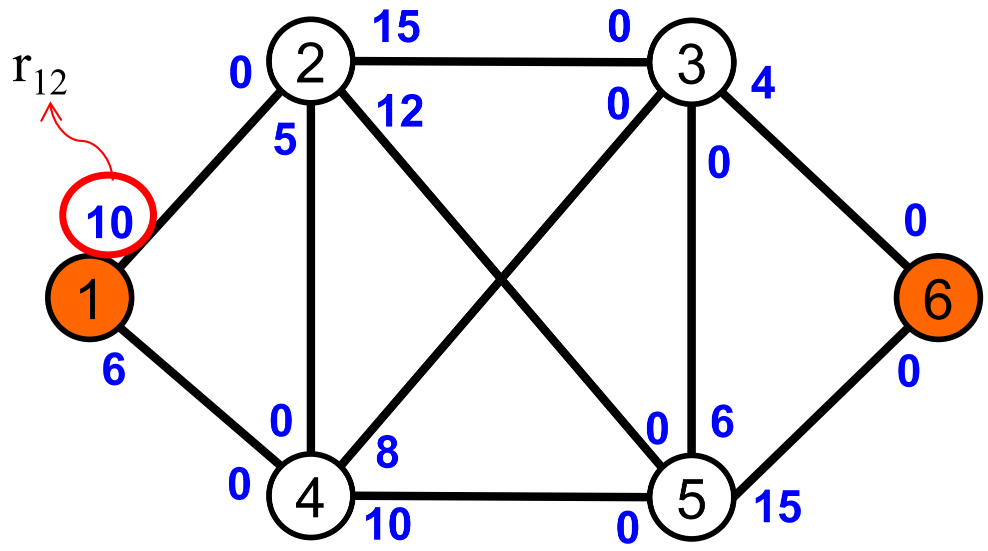


Calcoliamo la rete residua prodotta dal seguente flusso ammissibile x:

$$\underline{x}^T = \begin{bmatrix} x_{12} & x_{14} & x_{23} & x_{24} & x_{25} & x_{36} & x_{43} & x_{45} & x_{53} & x_{56} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 3 & 0 & 0 & 3 & 0 & 1 & 0 & 1 \end{bmatrix}$$

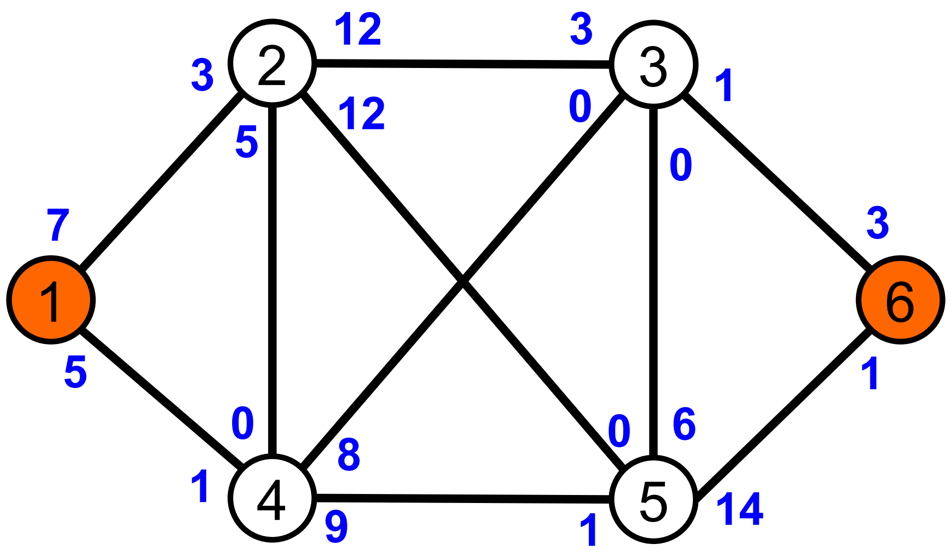
Grafo ausiliario e capacità residue





Calcoliamo la rete residua prodotta dal seguente flusso ammissibile $\underline{x}$:

$$\underline{x}^T = [x_{12}\; x_{14}\; x_{23}\; x_{24}\; x_{25}\; x_{36}\; x_{43}\; x_{45}\; x_{53}\; x_{56}] = [3\; 1\; 3\; 0\; 0\; 3\; 0\; 1\; 0\; 1]$$



r_{12}=10-3+0=7

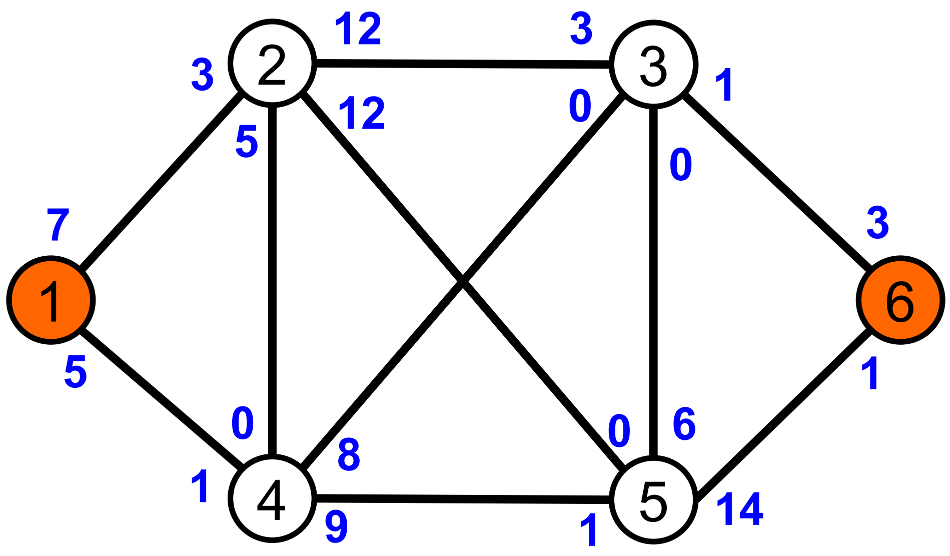
r_{21}=0-0+3=3

r_{23}=15-3+0=12

r_{32}=0-0+3=3

…

Grafo ausiliario e capacità residue



➢Se un arco ha capacità residua maggiore di zero, significa che posso spedire ancora del flusso attraverso quell'arco.

➢Se riesco ad individuare un cammino da *s* a *t* sul grafo ausiliario allora posso spedire del flusso addizionale dalla sorgente al pozzo.

➢Un cammino da *s* a *t* sul grafo ausiliario viene definito **cammino aumentante**.

➢Fino a quando nel grafo ausiliario sono presenti cammini aumentanti è sempre possibile incrementare il flusso da *s* a *t*.

Algoritmo dei Cammini Aumentanti

L'algoritmo dei cammini aumentanti risolve il problema del flusso massimo utilizzando il grafo ausiliario (o delle capacità residue) per stabilire come instradare il flusso sulla rete.

Consideriamo un grafo G=(V,A) ed un flusso ammissibile $\underline{x}$ (inizialmente il metodo considera il flusso nullo ossia $x_{ij}=0 \ \forall (i,j) \in A$.

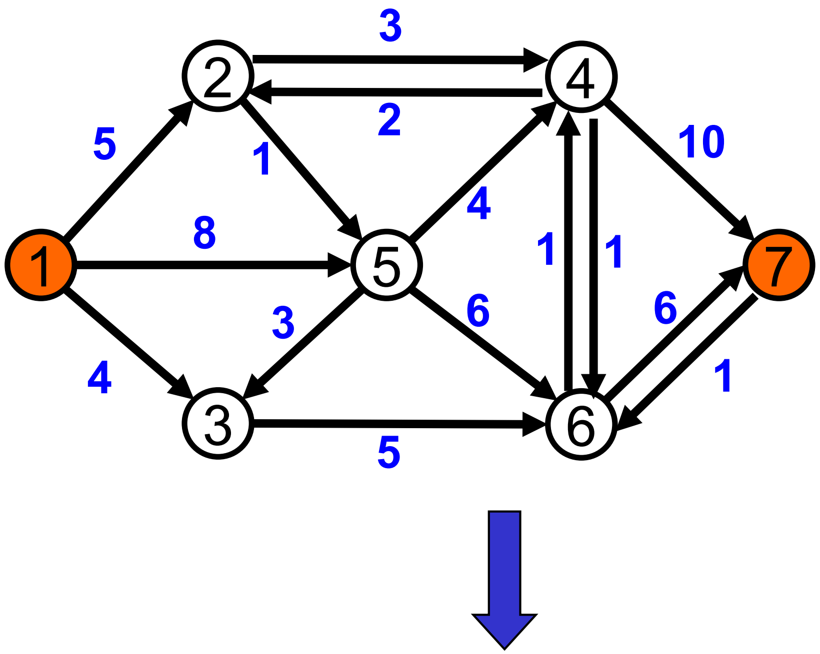
I passi principali dell'algoritmo dei cammini aumentanti sono:

1. Individuare nel grafo ausiliario un qualsiasi cammino p dal nodo sorgente al nodo pozzo su cui è possibile far transitare una quantità di flusso $\Delta>0$ (cammino aumentante). Se non esiste tale cammino, l'algoritmo si arresta.

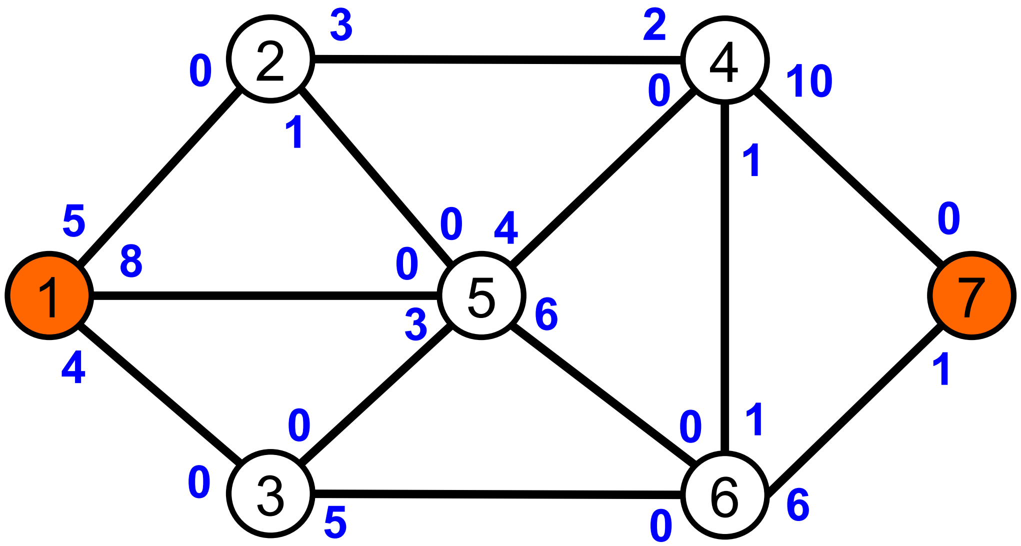
2. Il valore del flusso da inviare lungo il cammino p è pari alla capacità residua minima degli archi in p (i.e. $\Delta=\min\{r_{ij}:(i,j) \in p\}$)

3. Incrementare di Δ il valore del flusso f corrente, quindi $f=f+\Delta$, e aggiornare le capacità residue degli archi lungo il cammino p nel seguente modo:
   $r_{ij}= r_{ij} - \Delta$ e $r_{ji}= r_{ji} + \Delta$.

Algoritmo dei Cammini Aumentanti

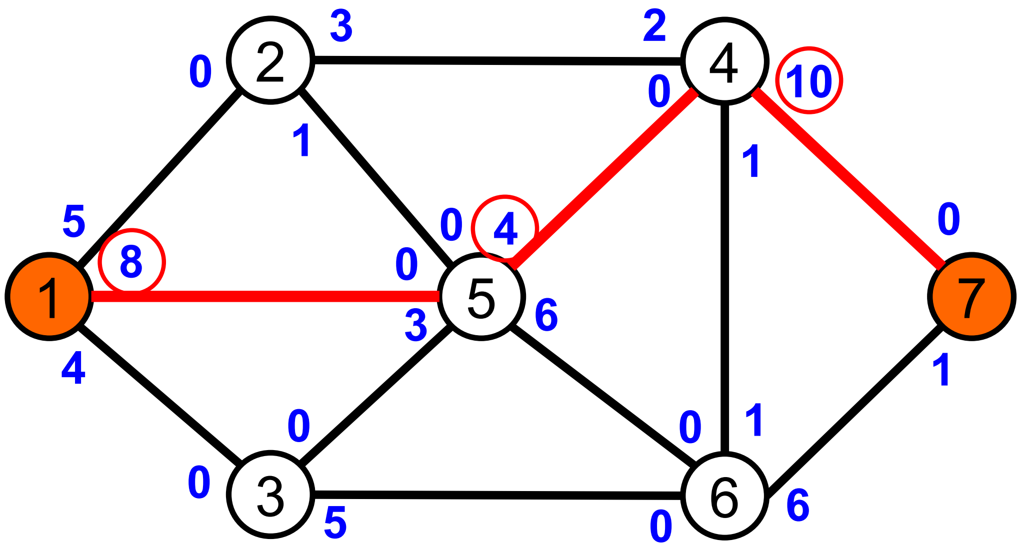


G(V,A) grafo in input
s=1 Nodo sorgente
t=7 nodo pozzo
$x_{ij}=0\ \ \forall(i,j)\in A$ flusso ammissibile iniziale



f = 0

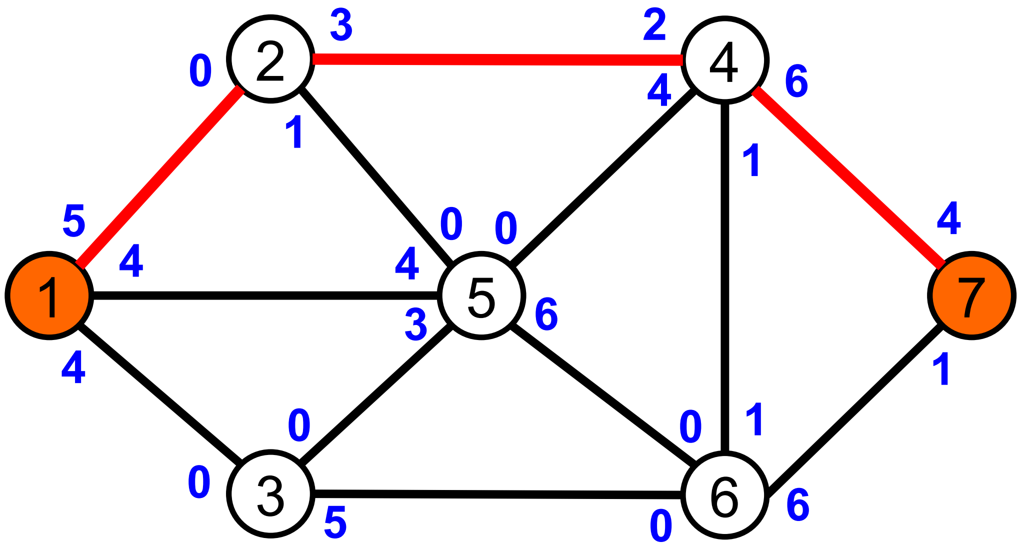
Algoritmo dei Cammini Aumentanti



Un path aumentante è un path da s a t sul grafo ausiliario.
Viene chiamato ''aumentante'' perché permette di aumentare il flusso sul grafo da s a t utilizzando gli archi del path.
Il flusso che posso spedire è uguale alla minima capacità residua degli archi del path.

P = 1-5-4-7

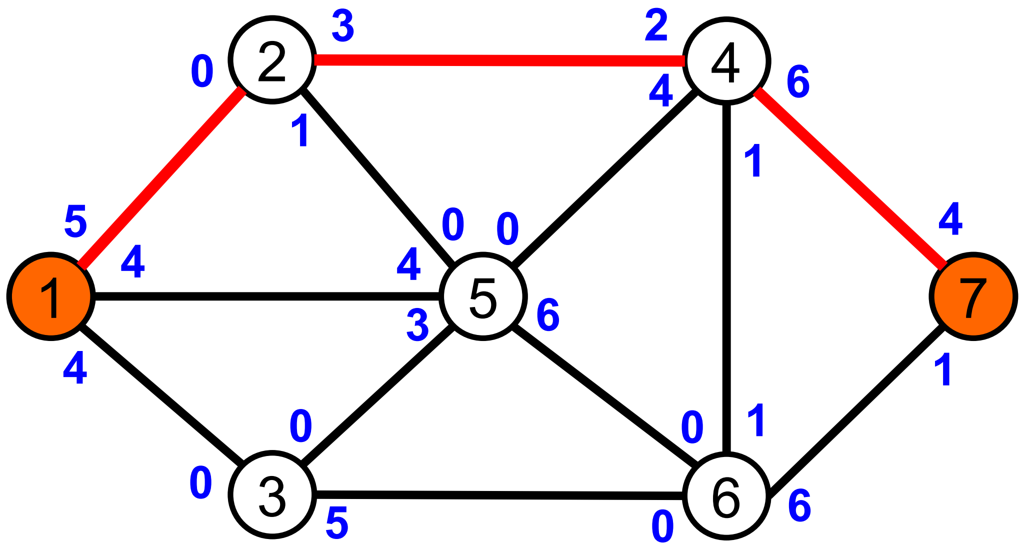
Δ=4 **f = 4**



P = 1-2-4-7

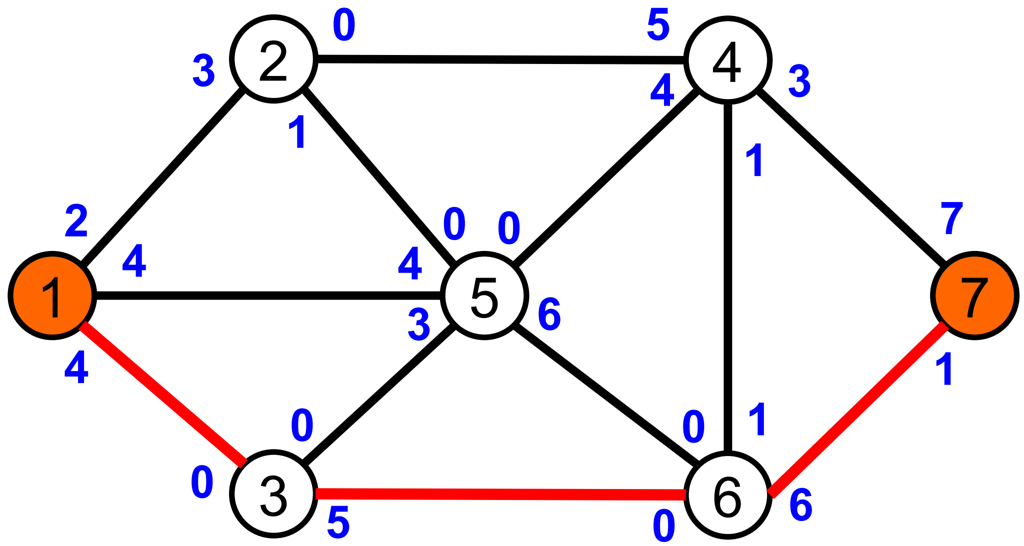
Δ=3 **f = f+Δ = 7**

Algoritmo dei Cammini Aumentanti



P = 1-2-4-7

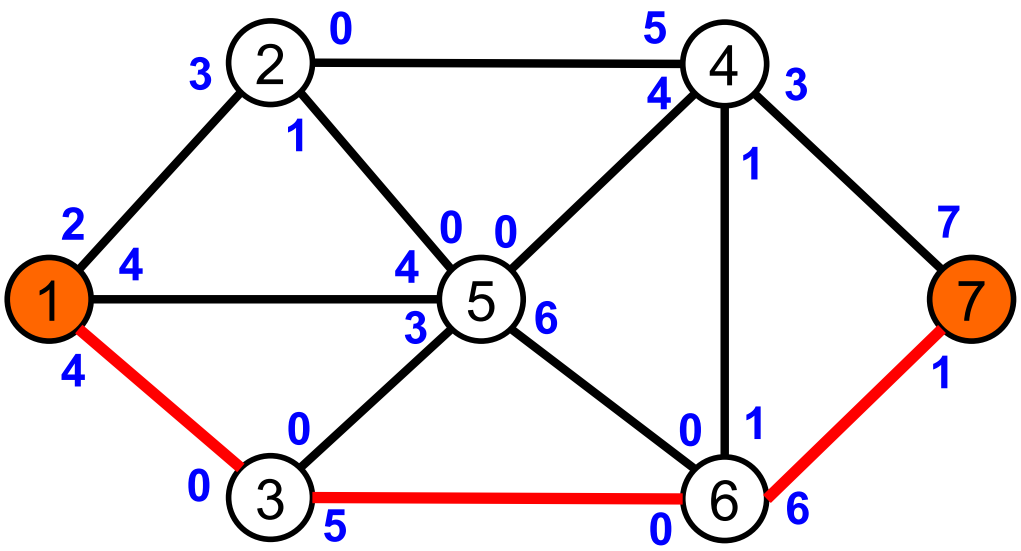
Δ=3 f = f+Δ = 7



P = 1-3-6-7

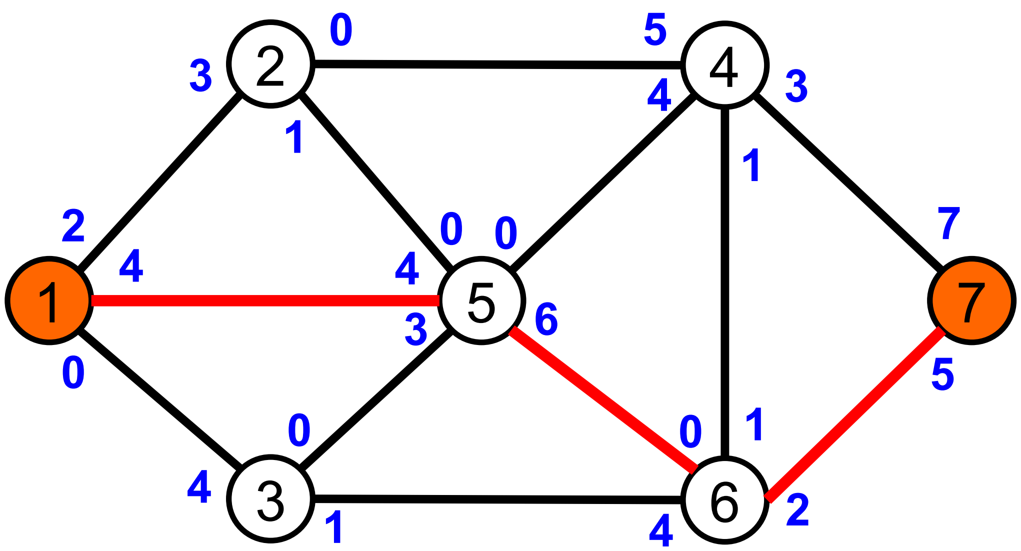
Δ=4 f = f+Δ = 11

Algoritmo dei Cammini Aumentanti



P = 1-3-6-7

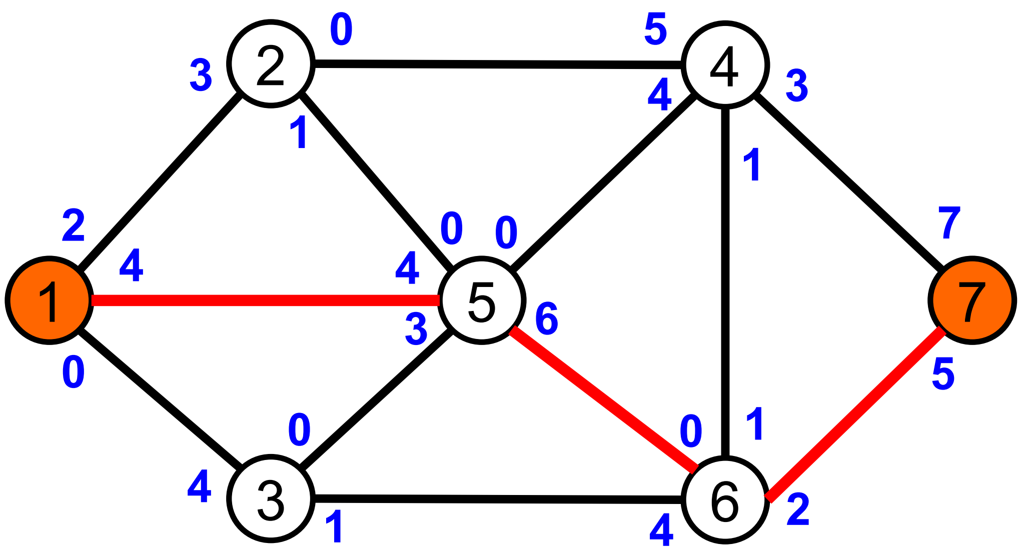
Δ=4 **f = f+Δ = 11**



P = 1-5-6-7

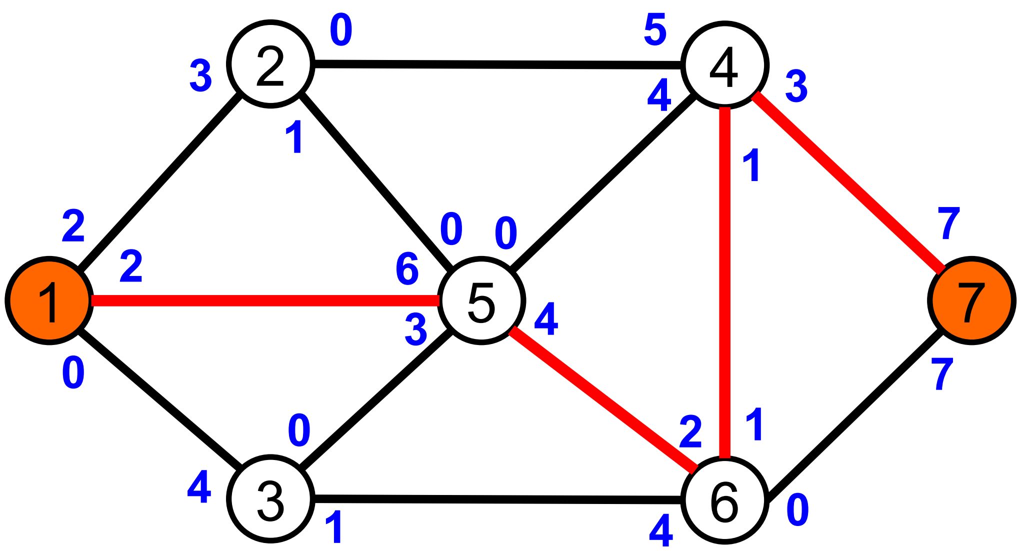
Δ=2 **f = f+Δ = 13**

Algoritmo dei Cammini Aumentanti



P = 1-5-6-7

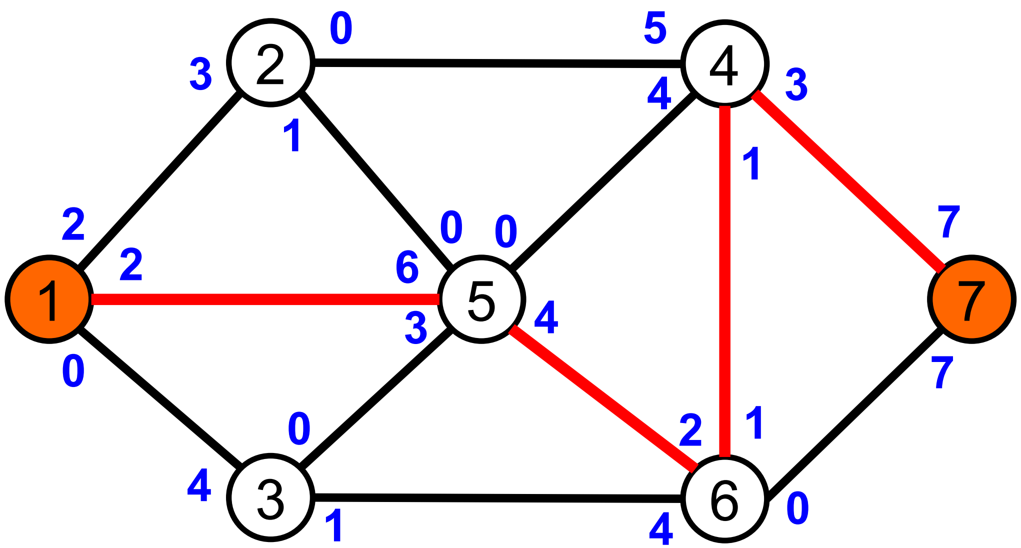
Δ=2 f = f+Δ = 13



P = 1-5-6-4-7

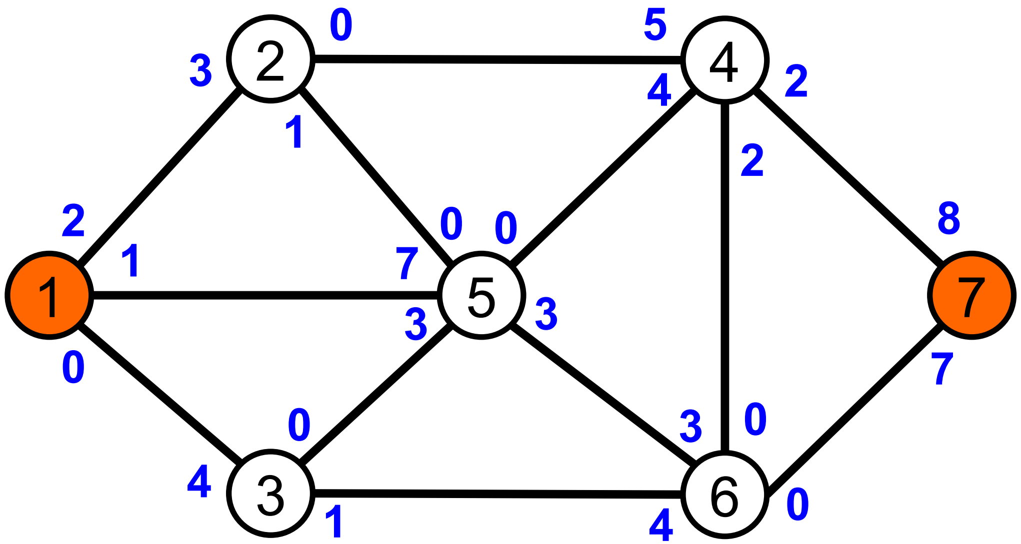
Δ=1 f = f+Δ = 14

Algoritmo dei Cammini Aumentanti



P = 1-5-6-4-7

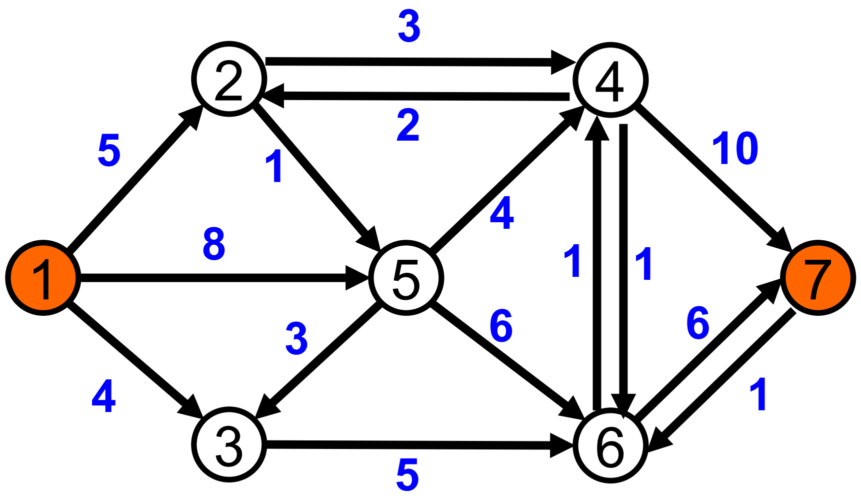
Δ=1 **f = f+Δ = 14**



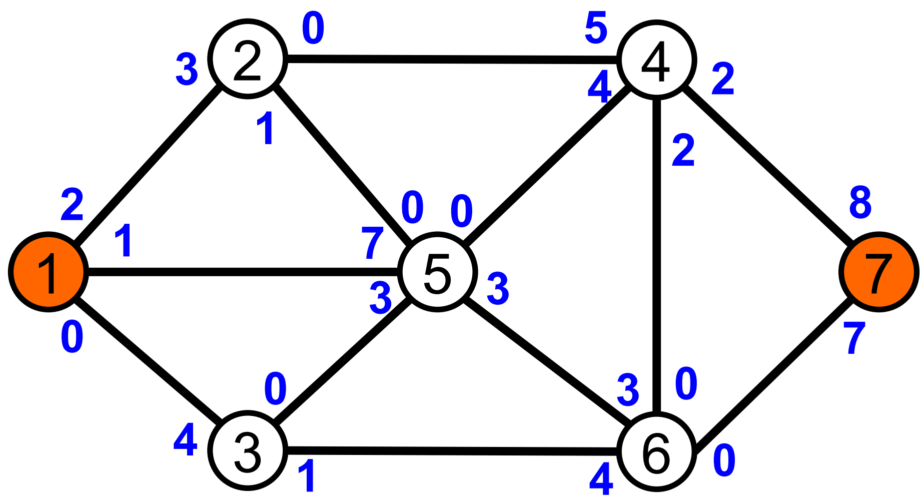
Non riesco ad individuare un cammino aumentante → Il flusso che ho individuato è ottimo

Algoritmo dei Cammini Aumentanti

Grafo iniziale



Grafo finale



Il valore delle variabili decisionali, per ogni arco del grafo di partenza, è pari alla differenza tra la capacità originale dell'arco meno quella residua nell'ultimo grafo ausiliario (se tale valore è negativo la variabile decisionale varrà zero). Formalmente: $\boldsymbol{x_{ij} = \max\{0 \, , u_{ij} - r_{ij}\}}$
Ad esempio per l'arco (1,2) abbiamo una capacità iniziale pari a 5 e una finale pari a 2 quindi $x_{12}=3$.

Analogamente abbiamo
$x_{15}=8-1=7$, $x_{13}=4-0=4$,
$x_{25}=1-1=0$, $x_{24}=3-0=3$ *ecc.ecc.*

Dettagli per la correttezza e l'implementazione

Si noti che nello schema generale del metodo dei cammini aumentanti ci sono dei dettagli che devono essere meglio chiariti:

- Come si individua un cammino aumentante o come si mostra che non esiste un cammino aumentante?

- Come certificare che il flusso ottenuto è quello massimo?

La risposta a queste domande puo' essere ottenuta considerando una particolare implementazione dell'algoritmo del cammino aumentante che da luogo al ***Labeling Algorithm di Ford and Fulkerson***

Labeling Algorithm di Ford and Fulkerson

Idea Principale:

➢Tramite opportuni algoritmi di visita è possibile etichettare tutti i nodi che possono essere raggiunti dalla sorgente nel grafo ausiliario tramite archi con capacità residua positiva;

➢Se il pozzo viene etichettato durante la precedente visita, allora esiste un cammino aumentante da *s* a *t* tramite il quale è possibile inviare ulteriori unità di flusso.;

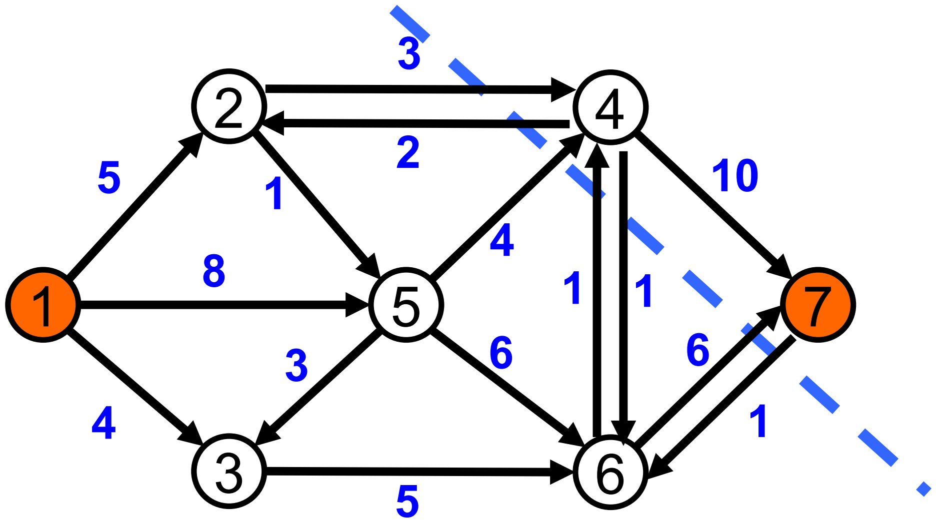
➢Se il pozzo non viene etichettato allora si costruisce un taglio nel seguente modo:

- in V_1 si inseriscono i nodi etichettati (ovvero raggiungibili da *s*)
- in V_2 si inseriscono i nodi non etichettati

➢ Poichè la capacità del taglio così costruito è pari al flusso f inviato fino a quel momento. Dal teorema MinCut/MaxFlow tale flusso f è massimo.

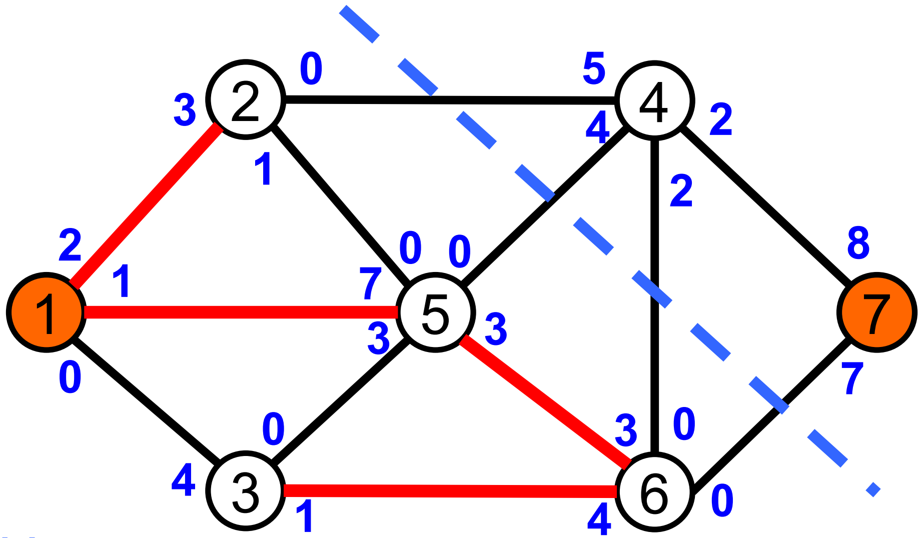
Individuazione taglio minimo

Grafo iniziale



Capacità = 14

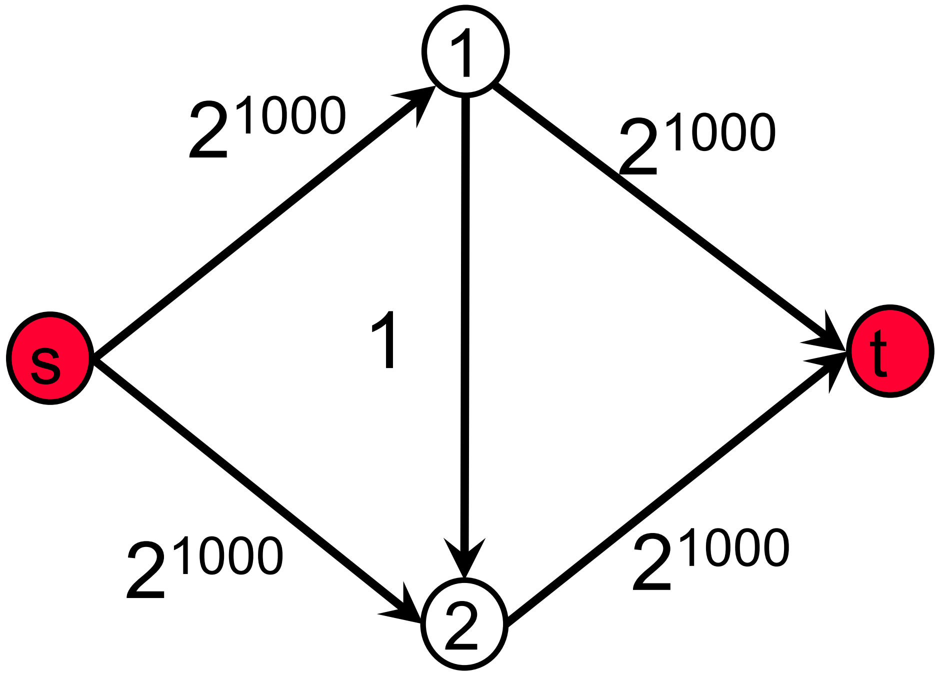
Grafo finale



Taglio : V_1 ={1,2,3,5,6}, V_2 = {4,7}

Per poter individuare il taglio minimo, la cui capacità sarà uguale al flusso massimo f=14, è sufficiente controllare quali sono i nodi raggiungibili dalla sorgente 1 attraverso archi con capacità residua >0 nell'ultimo grafo ausiliario.

Complessità dell'algoritmo del grafo ausiliario



Potremmo idealmente utilizzare in sequenza i cammini aumentanti s-1-2-t, s-2-1-t, s-1-2-t, s-2-1-t …

Approcci alternativi

Per migliorare la complessità dell’algoritmo ci sono diversi approcci:

- cercare un cammino con il numero minimo di archi (shortest augmenting path algorithm)

- posso cercare un cammino con una capacità almeno pari ad una quantità Δ fissata di volta in volta (capacity scaling algorithm)

- algoritmi di preflow push