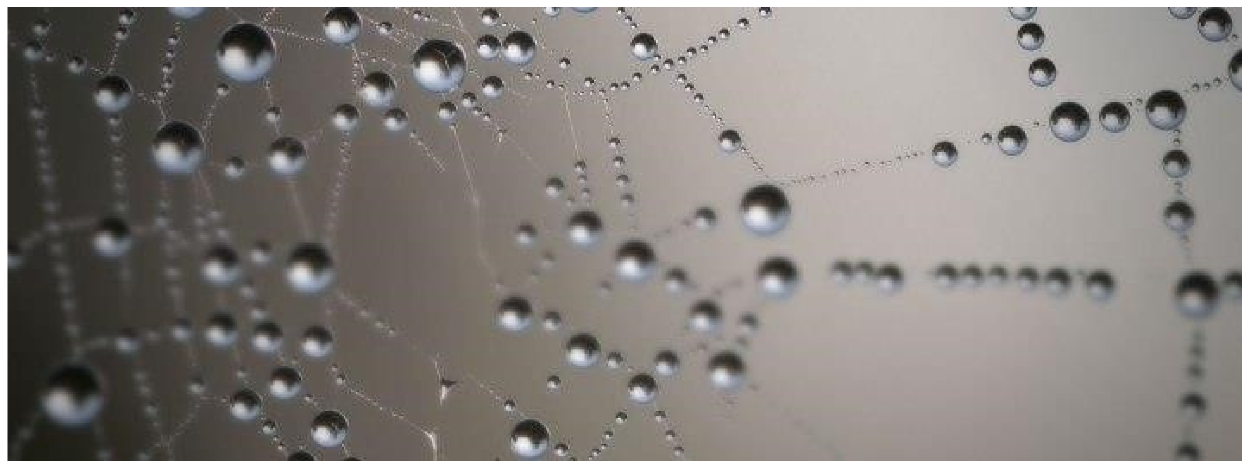
Università Politecnica delle Marche

Introduzione alla Teoria dei Grafi – parte II

ver 2.0.0



Fabrizio Marinelli
fabrizio.marinelli@staff.univpm.it
tel. 071 - 2204823

Sommario

- Insiemi indipendenti e coperture
- Relazioni tra le strutture
- algoritmo Greedy

[Ipotesi di lavoro] Grafi non orientati e connessi

Premessa: insiemi massimali e massimi

Sia U un insieme finito e discreto di elementi e $\wp$ un predicato che esprime una proprietà data.

Rispetto alla proprietà $\wp$, un insieme $S \subseteq U$ si dice:

- massimale, se ogni sottoinsieme Q di U che contiene propriamente S non soddisfa $\wp$.
- massimo, se ogni sottoinsieme Q di U che soddisfa $\wp$ non ha più elementi di S, cioè $|S| \geq |Q| \quad \forall Q \subseteq U$

Un insieme massimo è anche massimale
ma non vale in generale il viceversa

Esempio

$U = \{1, 3, 5, 4, 9, 7\}$

$\wp$ = « la somma è minore di 10 »

- $S = \{5, 4\}$ è un insieme massimale
- $S = \{7, 1\}$ è un insieme massimale
- $S = \{1, 3, 5\}$ è un insieme massimale e **massimo**
- $S = \{4, 3\}$ **non** è un insieme massimale (potrei aggiungere 1 e continuare a soddisfare $\wp$)

Premessa: insiemi minimali e minimi

Sia U un insieme finito e discreto di elementi e $\wp$ un predicato che esprime una proprietà data.

Rispetto alla proprietà $\wp$, un insieme $S \subseteq U$ si dice:

- minimale, se ogni sottoinsieme Q di U <u>contenuto propriamente</u> in S non soddisfa $\wp$.
- minimo, se ogni sottoinsieme Q di U che soddisfa $\wp$ <u>non ha meno elementi</u> di S, cioè $|S| \leq |Q| \quad \forall Q \subseteq U$

Un insieme minimo <u>è anche</u> minimale
ma non vale in generale il viceversa

Esempio

$U = \{1, 3, 5, 4, 9, 7\}$

$\wp$ = « la somma è maggiore di 10 »

- $S = \{5, 3, 4\}$ è un insieme minimale
- $S = \{1, 3, 7\}$ è un insieme minimale
- $S = \{9, 3\}$ è un insieme minimale e **minimo**
- $S = \{7, 3, 9\}$ **non** è un insieme minimale (potrei rimuovere 7 oppure 3 e continuare a soddisfare $\wp$)

Insiemi indipendenti

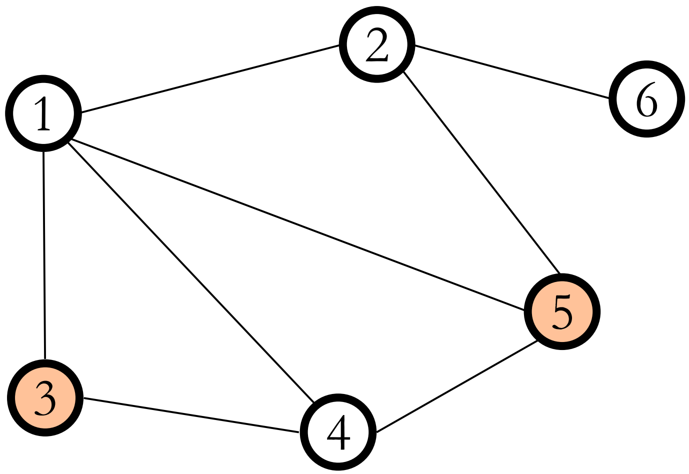
Dato un grafo non orientato $G = (V, E)$, un *insieme indipendente* è un qualsiasi sottoinsieme di nodi S (o di archi M) costituito da elementi mutuamente non adiacenti.

- S è detto *insieme stabile* (*stable set*)
- M è detto *abbinamento* (*matching*)

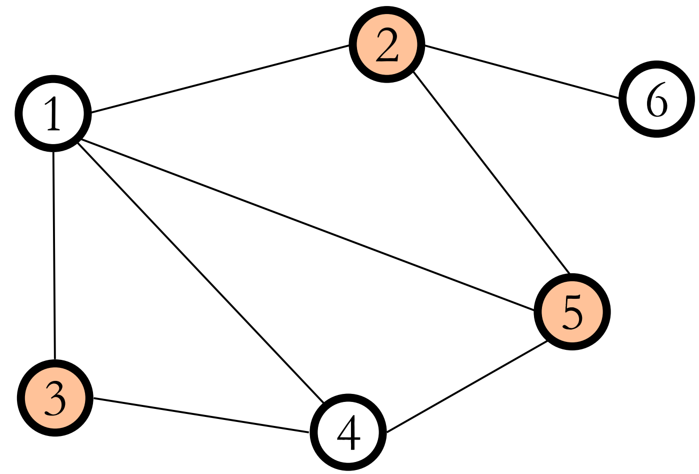
- Un insieme indipendente massimale di G non è contenuto propriamente in nessun insieme indipendente di G.
- Un insieme indipendente massimo è un insieme indipendente di cardinalità massima.

Insieme stabile

$S \subseteq V$ è un insieme stabile se $u,v \in S$ implica $\{u,v\} \notin E$.



$S = \{3,5\}$ è un insieme stabile

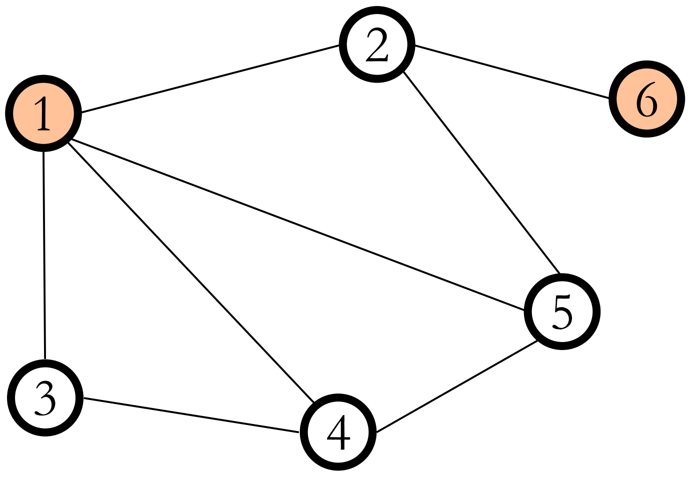


$S = \{2,3,5\}$ **non** è un insieme stabile (i nodi 2 e 5 sono adiacenti)

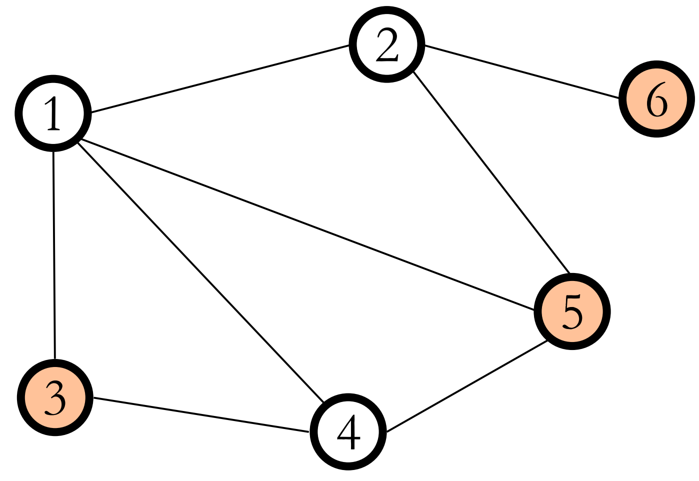
- $S = \varnothing$ è un insieme stabile.
- Ogni insieme costituito da un singolo nodo è un insieme stabile.

Insieme stabile

$S \subseteq V$ è un insieme stabile se $u,v \in S$ implica $\{u,v\} \notin E$.



insieme stabile massimale



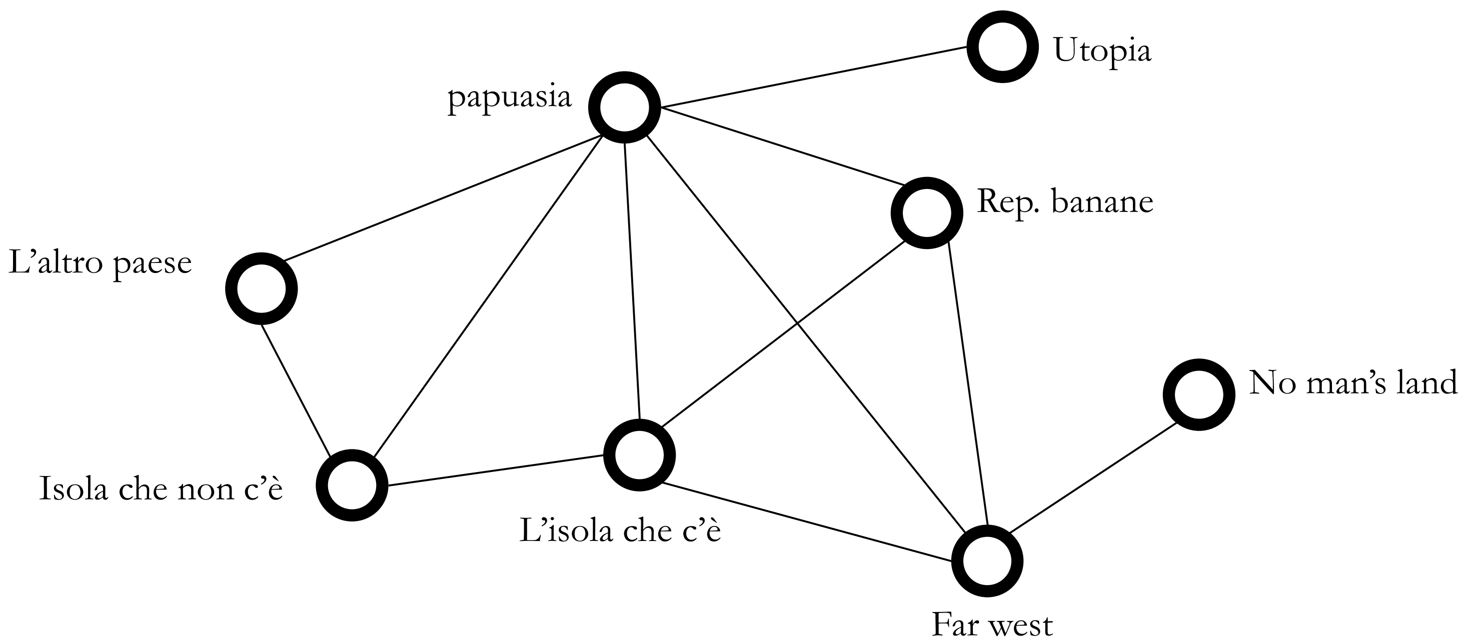
insieme stabile massimo

Applicazioni: war games (coalizioni)

Una grande potenza (…) vuole creare una grande coalizione per “sconfiggere il male”. Per avere successo però deve stare attenta a non coinvolgere Stati in conflitto reciproco. Quali sono le coalizioni «stabili»?

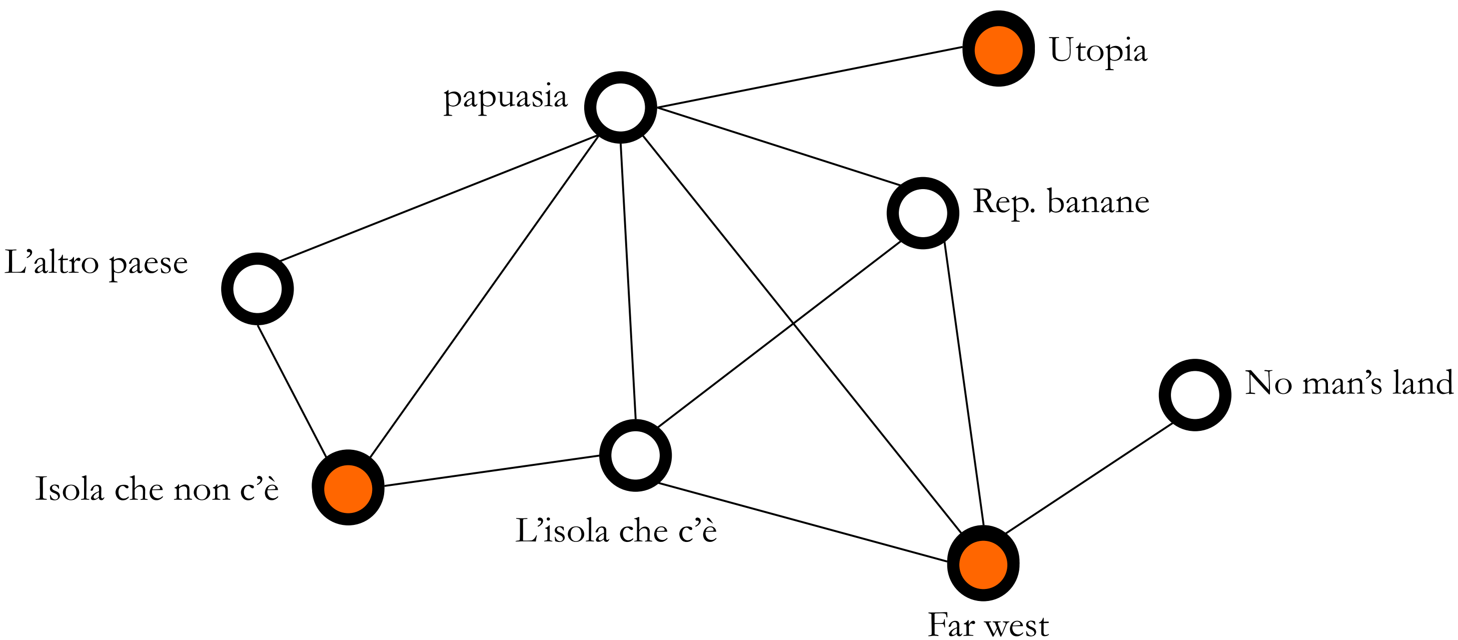
Una grande potenza (…) vuole creare una grande coalizione per "sconfiggere il male". Per avere successo però deve stare attenta a non coinvolgere Stati in conflitto reciproco. Quali sono le coalizioni «stabili»?

Definiamo un grafo $G = (V, E)$ in cui i nodi rappresentano gli stati e gli archi la relazione di conflitto, cioè esiste l'arco $\{u,v\}$ se gli stati u e v sono in conflitto.



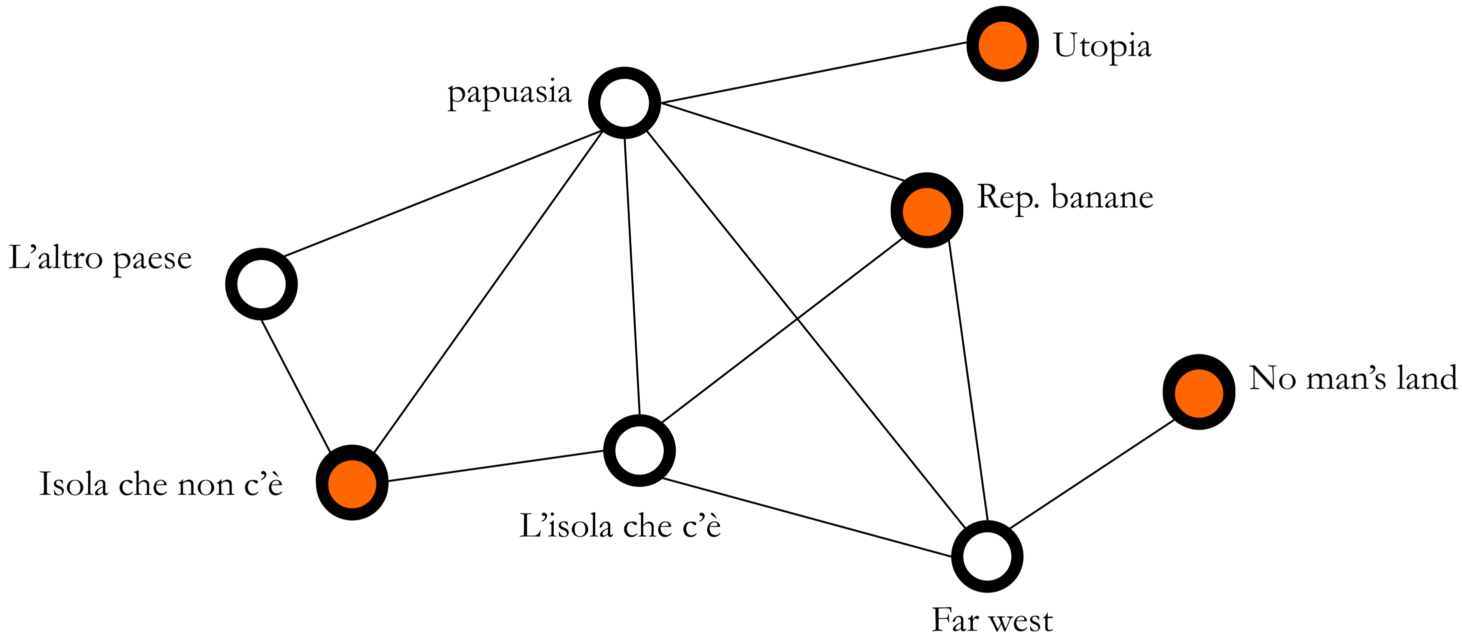
Un insieme stabile è un insieme di stati in pace tra loro: *Far West*, *Utopia* e *Isola che non c'è* possono allearsi perché non sono in conflitto reciproco.

La coalizione è massimale. Ma è anche massima?



…

Se *Utopia* e *Isola che non c'è* si alleano con *no man's land* e *Rep. delle banane* costituiscono un'alleanza più grande.
E' la più grande possibile? Esistono altre alleanze di 4 stati?



Applicazioni: insieme stabile (2)

Un manager deve assegnare un insieme di progetti a un team di ingegneri. In base alle competenze e alle compatibilità attitudinali ogni progetto deve essere svolto da un dato gruppo di ingegneri ma ogni ingegnere può eseguire un solo progetto. Quali sono i progetti che possono essere svolti contemporaneamente?

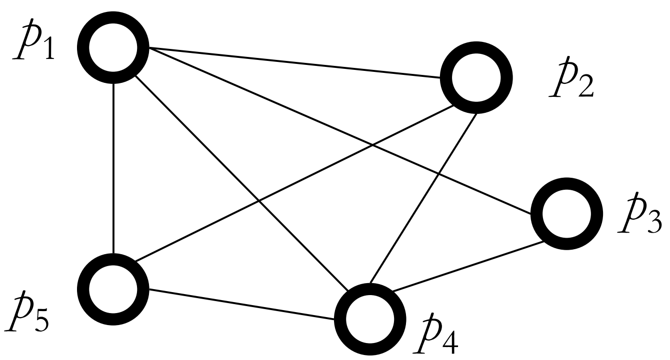
griglia di assegnamento

	p_1	p_2	p_3	p_4	p_5
Claudio	●		●	●	
Gino	●	●		●	●
Luca			●	●	
Andrea		●			

Applicazioni: insieme stabile (2) …

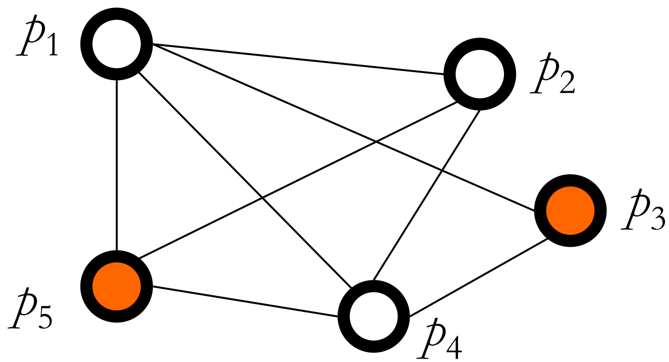
Definiamo un grafo $G = (V, E)$ in cui i nodi rappresentano i progetti e gli archi le incompatibilità tra progetti dovute alla richiesta dello stesso ingegnere, cioè esiste l'arco $\{u,v\}$ se i progetti u e v richiedono almeno un ingegnere in comune.

	p_1	p_2	p_3	p_4	p_5
Claudio	●		●	●	
Gino	●	●		●	●
Luca			●	●	
Andrea		●			



Applicazioni: insieme stabile (2) …

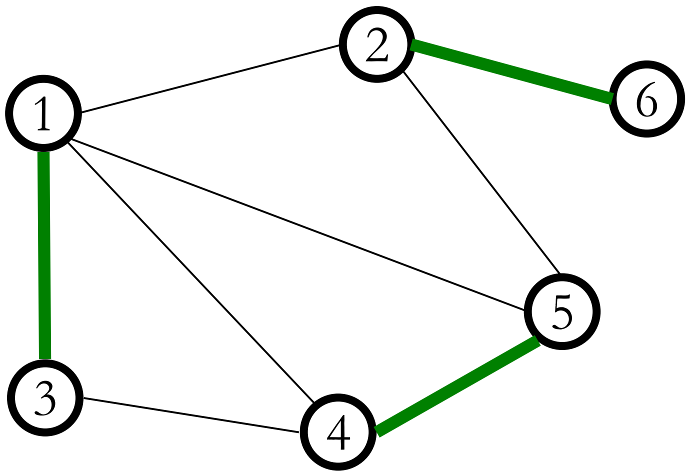
Un insieme stabile corrisponde a un insieme di progetti che possono essere svolti contemporaneamente.



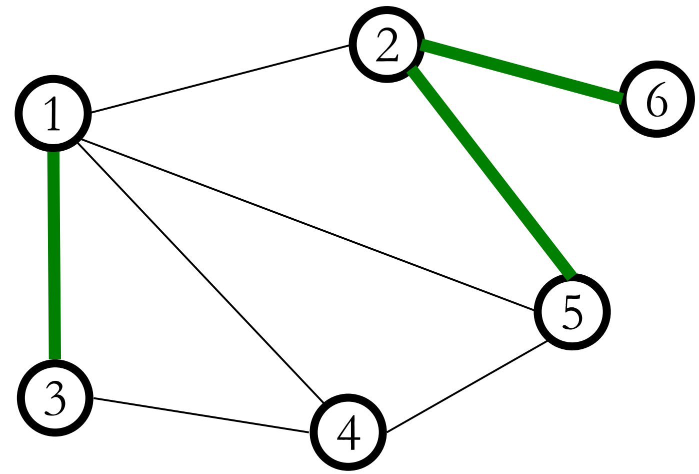
	p_1	p_2	p_3	p_4	p_5
Claudio	●		●	●	
Gino	~~●~~	~~●~~		~~●~~	●
Luca			●	●	
Andrea		●			

Abbinamento

$M \subseteq E$ è un abbinamento se uv, $hw \in M$ implica $u \neq v \neq h \neq w$.



$M = \{13,26,45\}$ è un abbinamento



$M = \{13,26,25\}$ **non** è un abbinamento (gli archi 25 e 26 sono adiacenti)

- $M = \varnothing$ è un abbinamento.
- Ogni insieme costituito da un singolo arco è un abbinamento.

Abbinamento

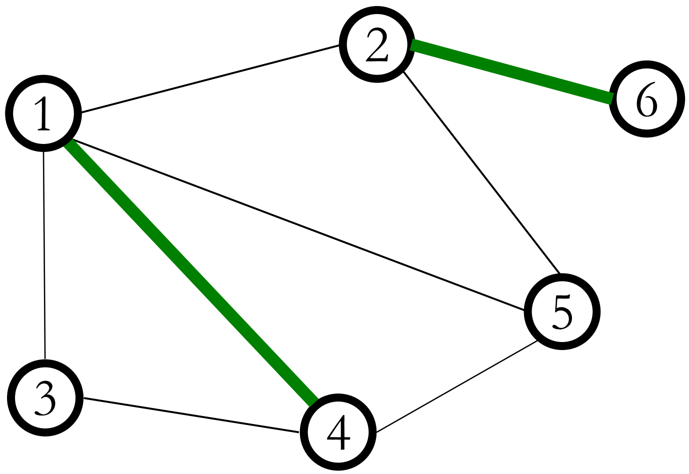
- Se G è un grafo bipartito allora anche un abbinamento M su G viene detto bipartito

- Se la cardinalità di un abbinamento M è $|V| / 2$ (cioè se ogni nodo di G è incidente su un arco di M) allora l'abbinamento viene detto perfetto

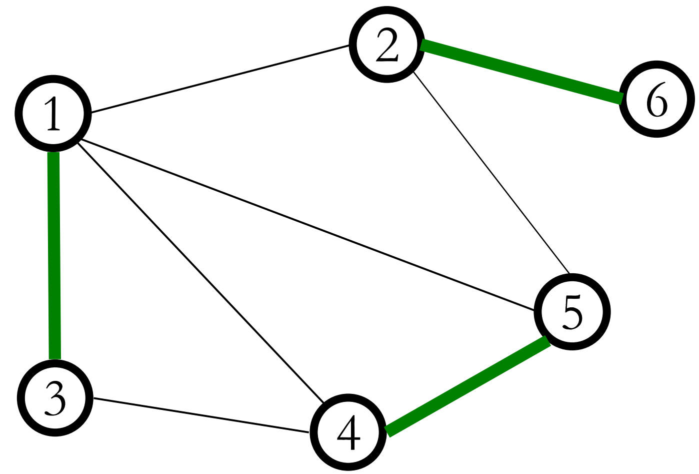
Ogni grafo ammette un abbinamento,
ma non è detto che esista un abbinamento perfetto

Abbinamento

$M \subseteq E$ è un abbinamento se uv, $hw \in M$ implica $u \neq v \neq h \neq w$.



Abbinamento massimale



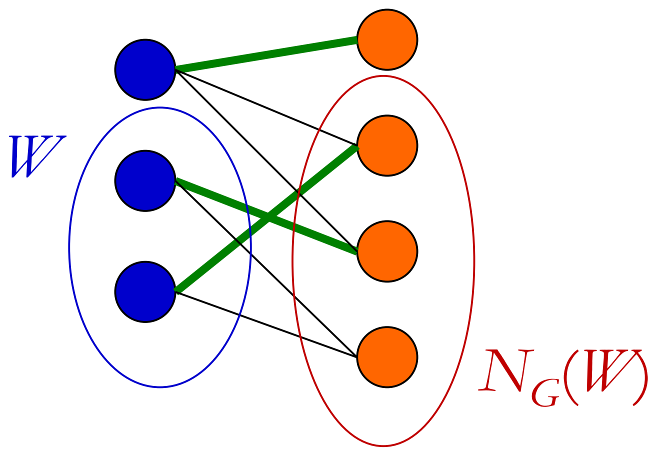
Abbinamento massimo

… e anche abbinamento perfetto

Abbinamento su grafi bipartiti

Teorema di Hall (marriage theorem): Sia $G = (V_1 \cup V_2, E)$ un grafo bipartito. G ammette un abbinamento che copre interamente V_1 **se e solo se** per ogni sottoinsieme W di V_1 si ha

$$| W | \leq | N_G(W) |$$



Corollario: se $| V_1 | = | V_2 |$ allora G ammette un abbinamento perfetto se e solo se valgono le condizioni del marriage theorem

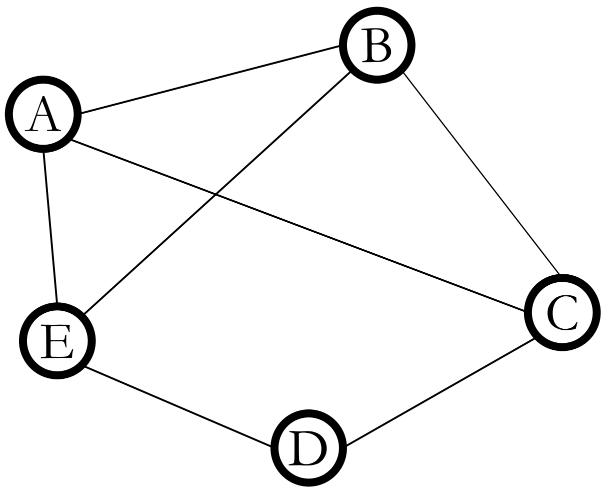
Applicazioni: abbinamento (1)

The *Fletcher Memorial Home*: una casa di riposo offre camere doppie a un gruppo di signori anziani. Ogni nonnetto vorrebbe condividere la stanza con uno dei suoi vecchi amici o, se non è possibile, stare da solo.

Quali sono gli accoppiamenti che accontentano tutti?

Vecchie amicizie	*Andrea*	*Bruno*	*Claudio*	*Daniele*	*Enzo*
Andrea		●	●		●
Bruno	●		●		●
Claudio	●	●		●	
Daniele			●		●
Enzo	●	●		●	

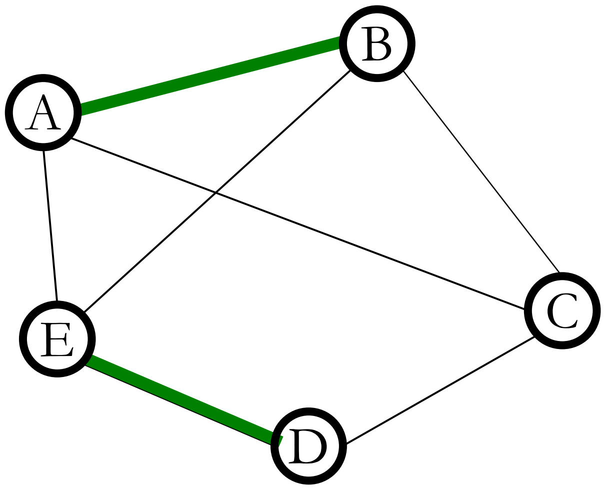
Applicazioni: abbinamento (1) …



nodi: nonnetti
archi: vecchie amicizie

Vecchie amicizie	*Andrea*	*Bruno*	*Claudio*	*Daniele*	*Enzo*
Andrea		●	●		●
Bruno	●		●		●
Claudio	●	●		●	
Daniele			●		●
Enzo	●	●		●	

Applicazioni: abbinamento (1) …



- L' abbinamento $M = \{\{A, B\},\{E, D\}\}$ descrive una possibile sistemazione dei nonnetti nelle stanze che rispetti i loro desideri.
- In particolare, l'arco $\{u, v\}$ dell'abbinamento descrive la stanza assegnata ai nonnetti u e v.
- Il grafo non ammette abbinamenti perfetti, quindi qualcuno dovrà dormire solo.

Applicazioni: abbinamento (2)

The *Fletcher Memorial Hospital* : una clinica specializzata in trapianti riceve *n* pazienti e dispone di *n* donatori. Siccome ogni donatore è compatibile con un certo gruppo di pazienti, ci si chiede:

E' possibile effettuare tutti i trapianti?

Pazienti

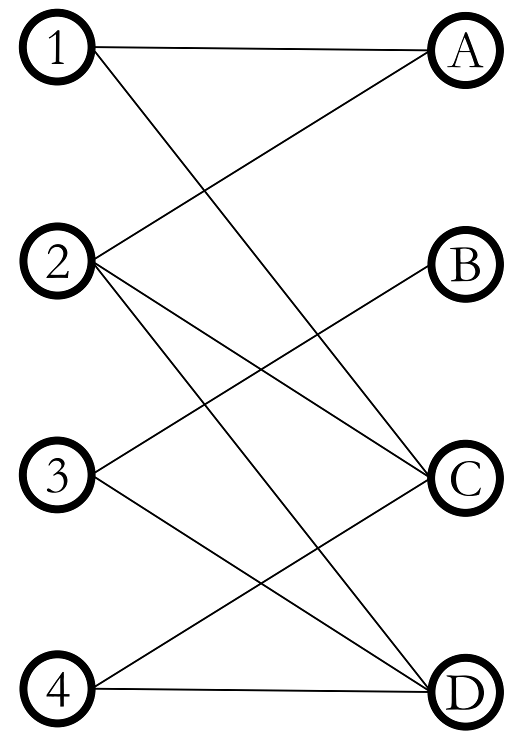
compatibilità	*A*	*B*	*C*	*D*
Donatore 1	●		●	
Donatore 2	●		●	●
Donatore 3		●		●
Donatore 4			●	●

Applicazioni: abbinamento (2) …

nodi: donatori e pazienti
archi: compatibilità

Pazienti

compatibilità	*A*	*B*	*C*	*D*
Donatore 1	●		●	
Donatore 2	●		●	●
Donatore 3		●		●
Donatore 4			●	●

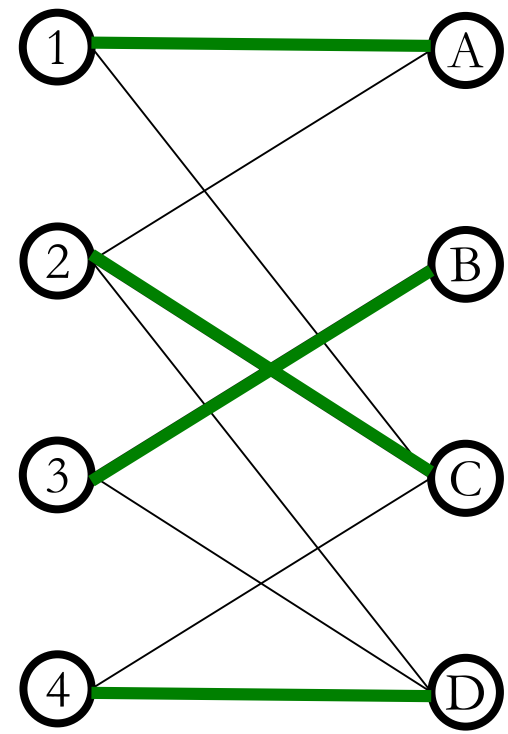


Applicazioni: abbinamento (2) …

- un abbinamento perfetto (se esiste) descrive un modo per effettuare <u>tutti</u> i trapianti.
- In particolare, l'arco $\{u, v\}$ dell'abbinamento indica la coppia u,v di donatore – paziente

Pazienti

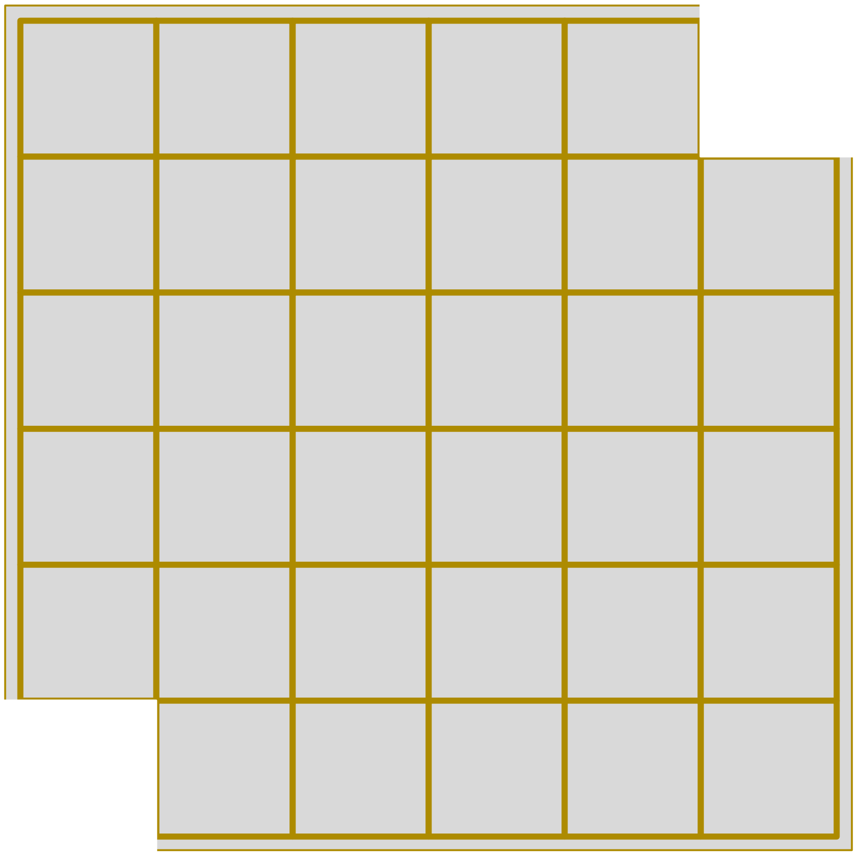
compatibilità	*A*	*B*	*C*	*D*
Donatore 1	●		●	
Donatore 2	●		●	●
Donatore 3		●		●
Donatore 4			●	●

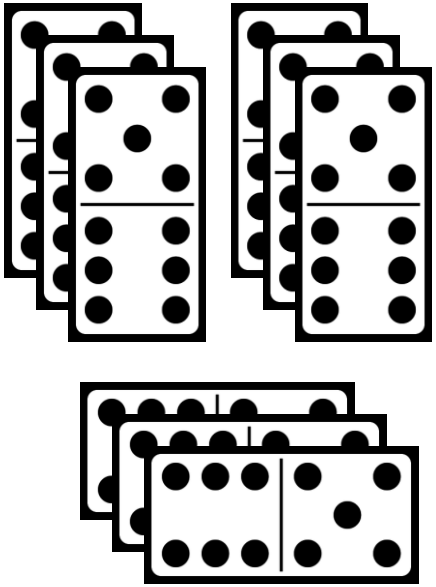


altri esempi…

… ancora con 'ste scacchiere!

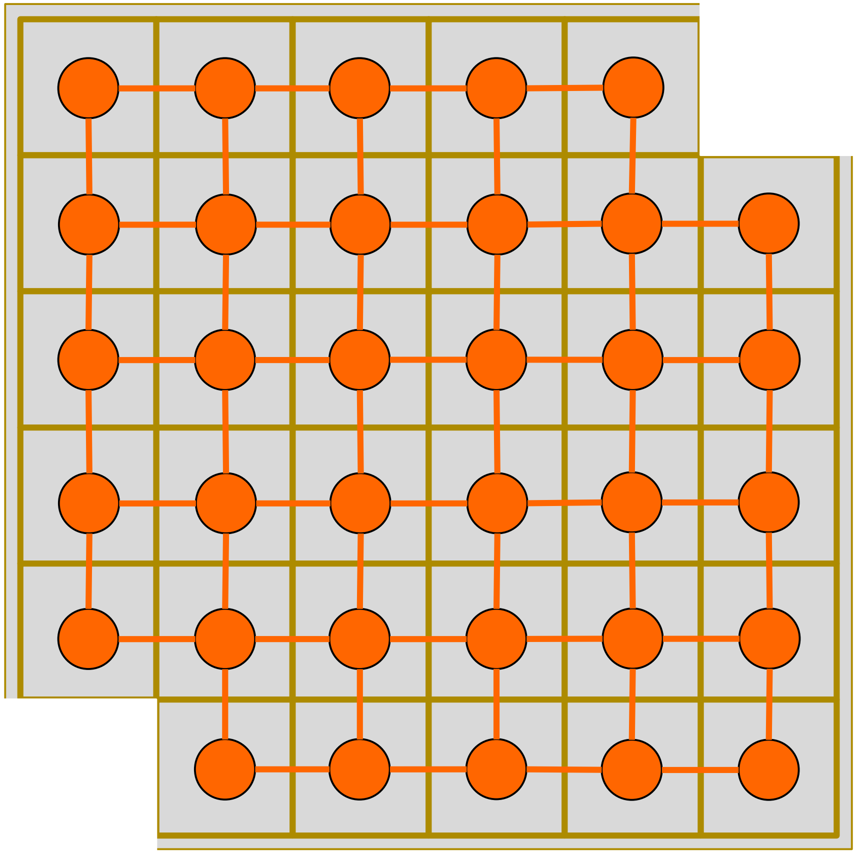
Disporre le tessere del domino (senza sovrapporle) in modo da coprire tutte le caselle





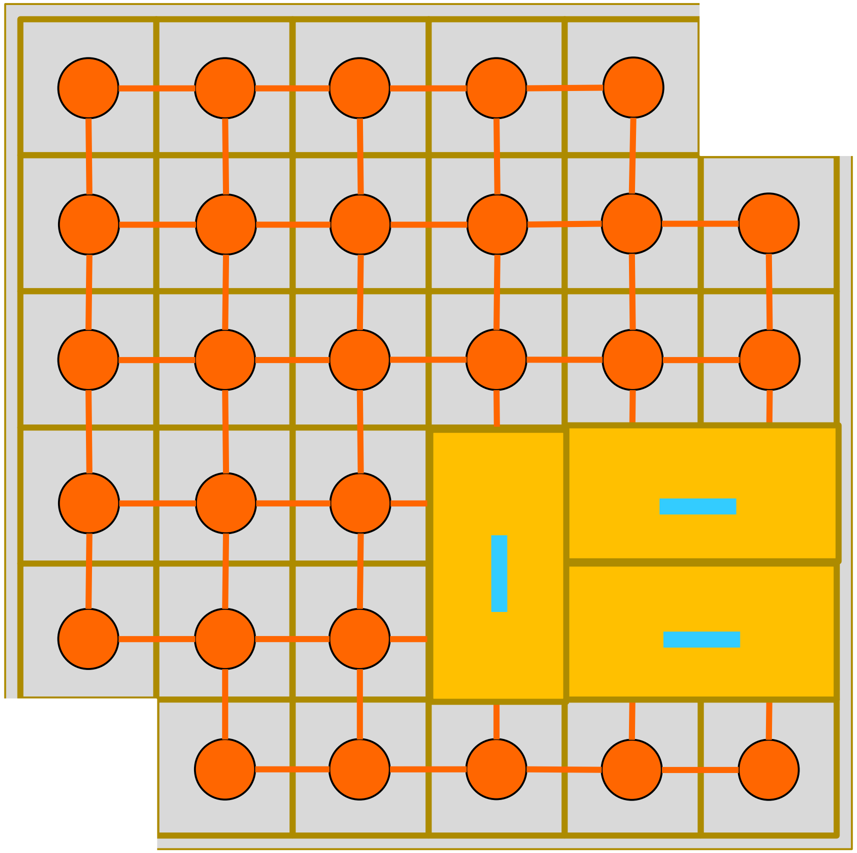
abbinamenti e dimostrazioni matematiche

associamo alla griglia un *grafo* che ha un nodo per ogni casella e in cui due nodi sono adiacenti se lo sono le corrispondenti caselle



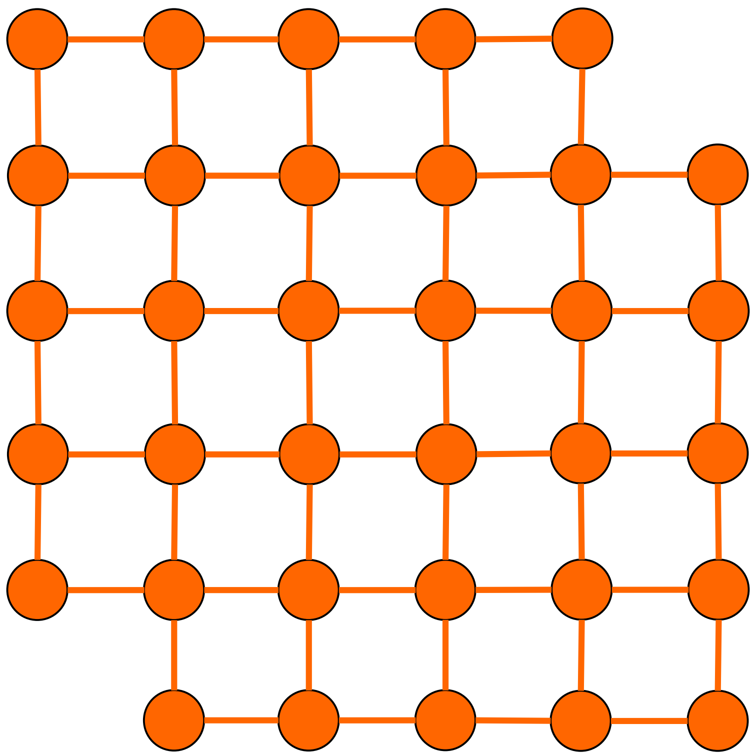
abbinamenti e dimostrazioni matematiche

- Posizionare una tessera equivale a selezionare un arco del grafo.
- Per evitare la sovrapposizione delle tessere, gli archi selezionati **non devono** avere nodi in comune (gli archi devono formare un *matching*)



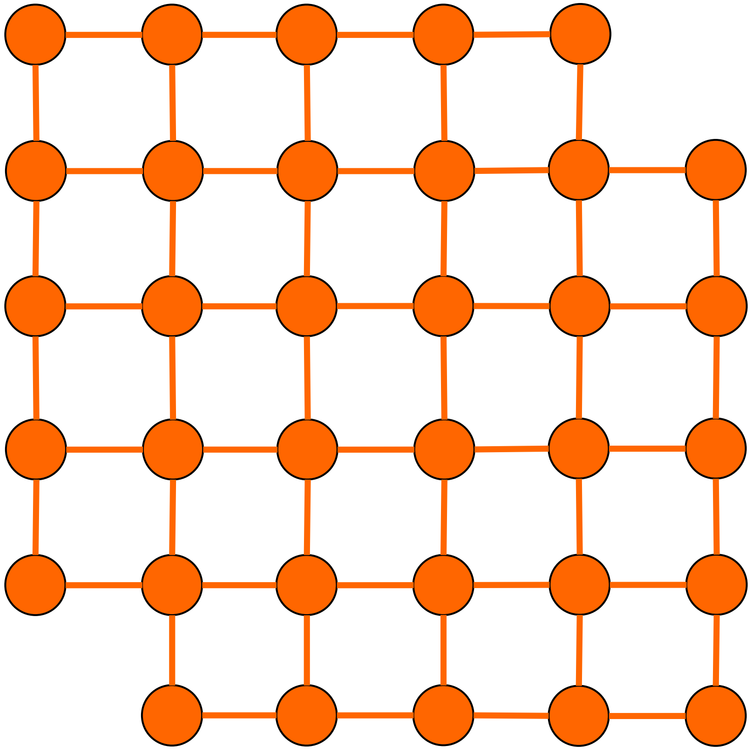
abbinamenti e dimostrazioni matematiche

- Siccome voglio coprire completamente la griglia, il matching deve essere *perfetto* (tutti i nodi devono essere «toccati» da archi del matching)



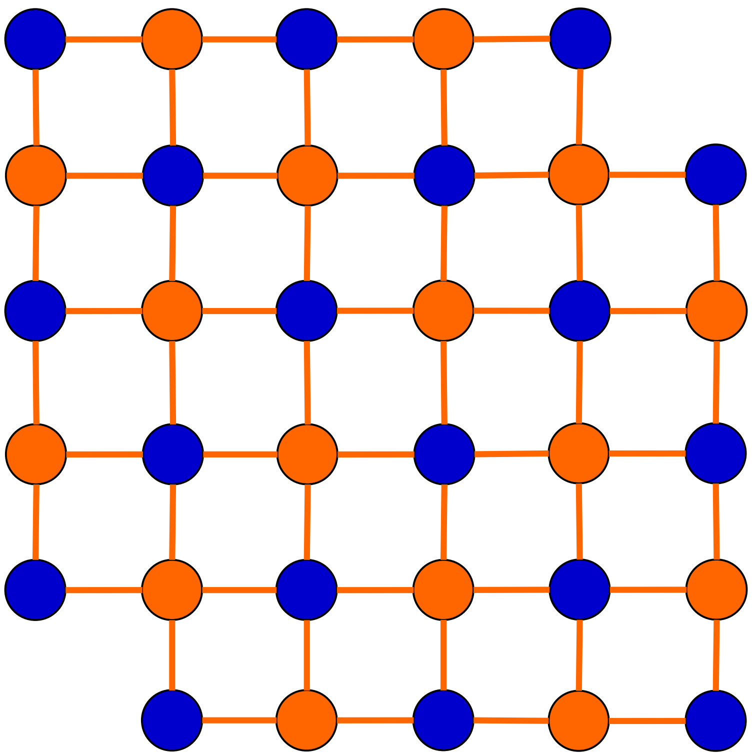
abbinamenti e dimostrazioni matematiche

- Il grafo è *bipartito*



abbinamenti e dimostrazioni matematiche

- teorema di Hall: il grafo ammette un matching che copre completamente i nodi blu **se e solo se** per ogni sottoinsieme W di nodi blu $|W| < |N(W)|$



16 nodi rossi e 18 nodi blu

Non esiste alcun matching perfetto

Copertura

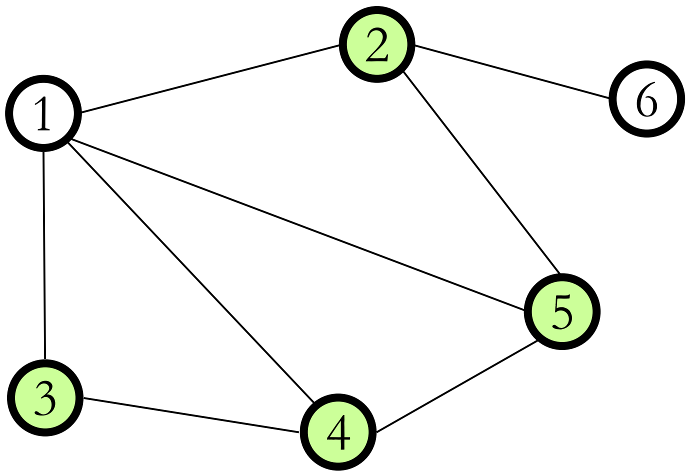
[Definizione] Dato un grafo non orientato $G = (V, E)$, una *copertura* è un qualsiasi sottoinsieme di nodi T (di archi F) tale che ogni arco (nodo) di G incida su almeno un elemento di T (di F).

- T è detto *copertura con nodi* (*trasversale* o *vertex-cover*)
- F è detto *copertuta con archi* (*edge-cover*)

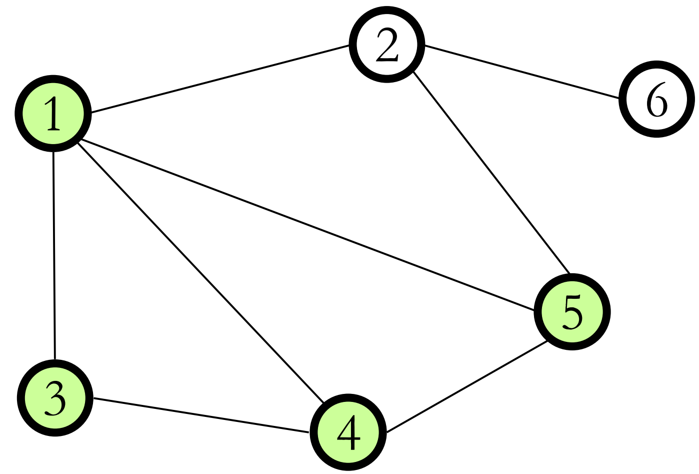
- Una copertura minimale di G non contiene propriamente alcuna copertura.
- Una copertura minima di G è una copertura di cardinalità minima

Copertura con nodi

$T \subseteq V$ è un trasversale se $\forall uv \in E \;\; u \in T$ oppure $v \in T$.



$T = \{2,3,4,5\}$ è un trasversale

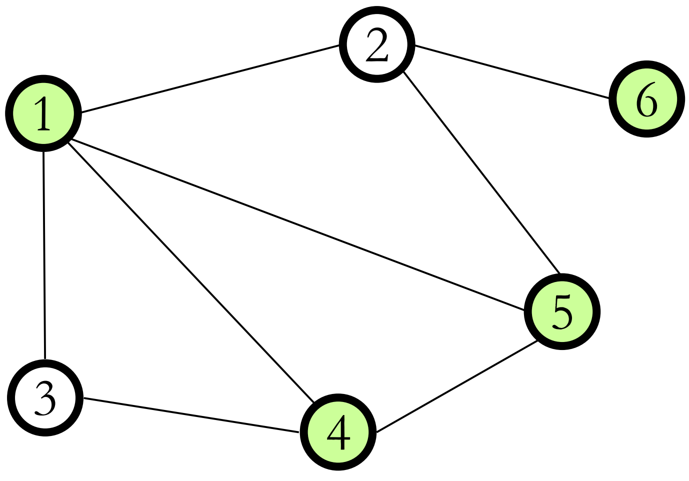


$T = \{1,3,4,5\}$ **non** è un trasversale (l'arco 26 non è «coperto»)

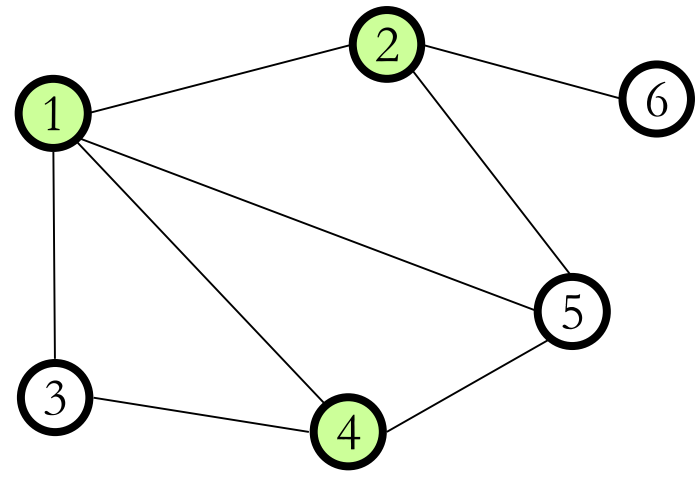
- $T = V$ è un trasversale (banale).

Copertura con nodi

$T \subseteq V$ è un trasversale se $\forall uv \in E \;\; u \in T$ oppure $v \in T$.



Trasversale minimale



trasversale minimo

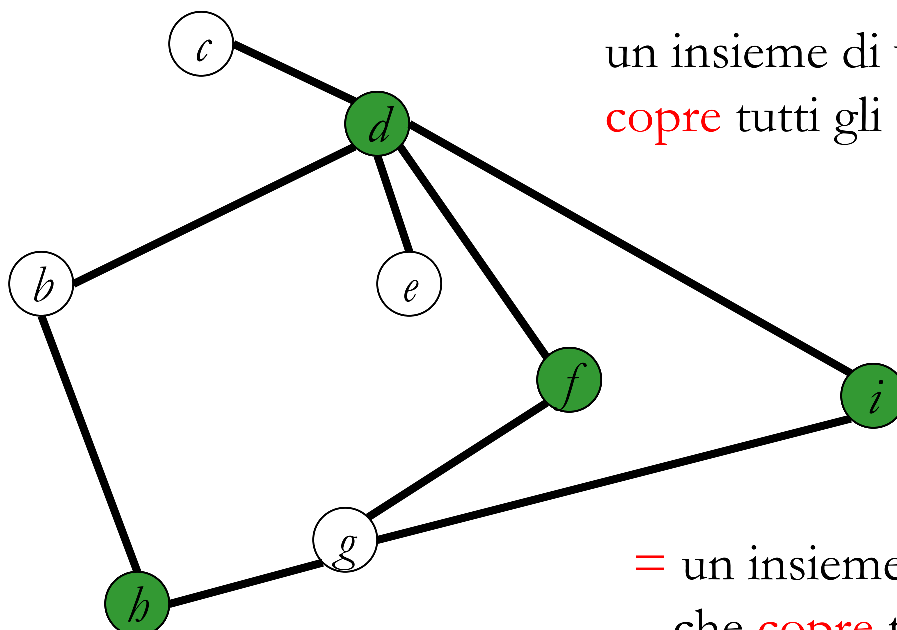
Applicazioni: copertura con nodi

Per monitorare una rete telematica si vuole individuare un insieme di server che accedano direttamente a tutti i link della rete. Qual è un insieme di server che soddisfa questo requisito?

Applicazioni: copertura con nodi …

Per monitorare una rete telematica si vuole individuare un insieme di server che accedano direttamente a tutti i link della rete. Qual è un insieme di server che soddisfa questo requisito?

Si può definire in modo naturale un grafo in cui i vertici rappresentano i server e gli archi i link diretti tra server.

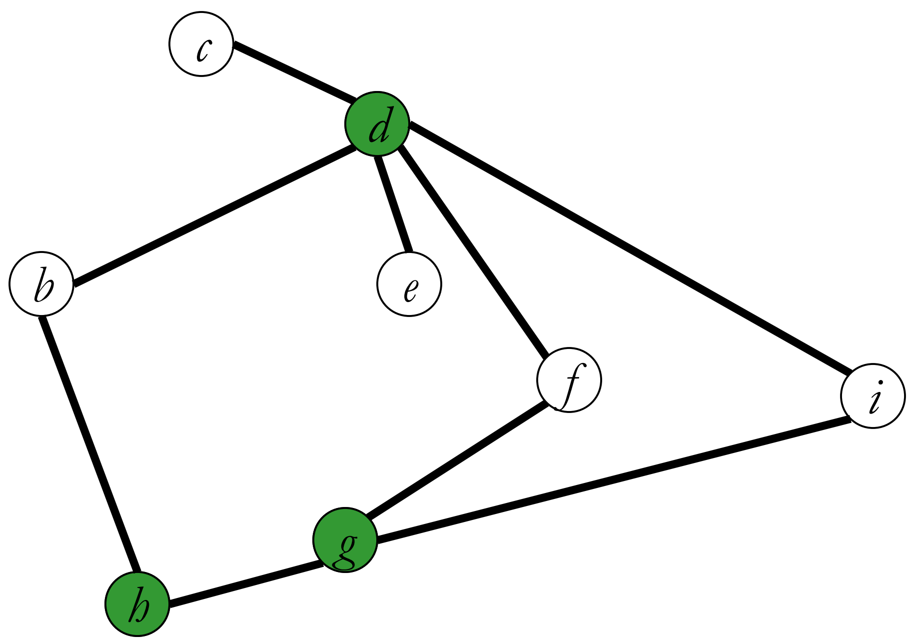


un insieme di vertici che copre tutti gli archi

= un insieme di server che copre tutti i link

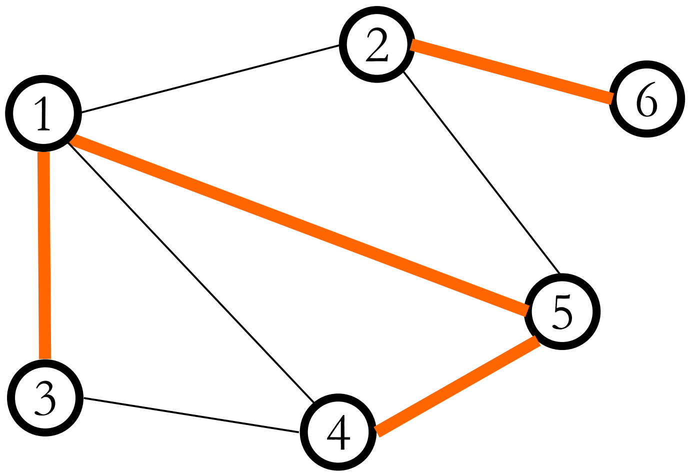
Applicazioni: copertura con nodi …

Una copertura migliore è l'insieme di nodi $\{d, h, g\}$.

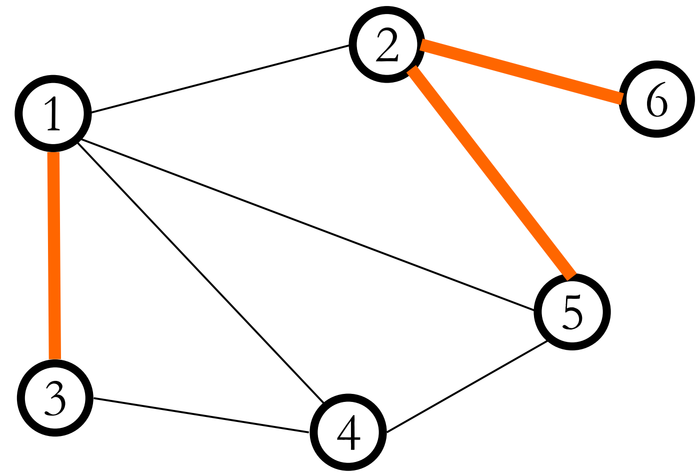


Copertura con archi

$F \subseteq E$ è una copertura con archi se $\forall u \in V \ \exists \ uv \in F$.



$F = \{13,15,26,45\}$ è una copertura con archi

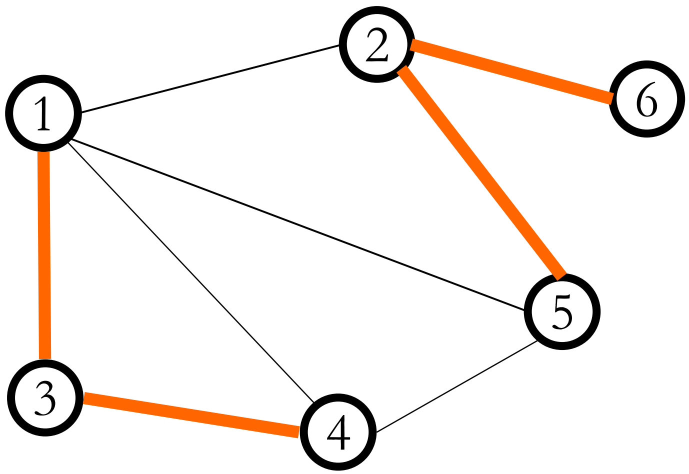


$F = \{13,26,25\}$ **non** è una copertura con archi (il nodo 4 non è «coperto»)

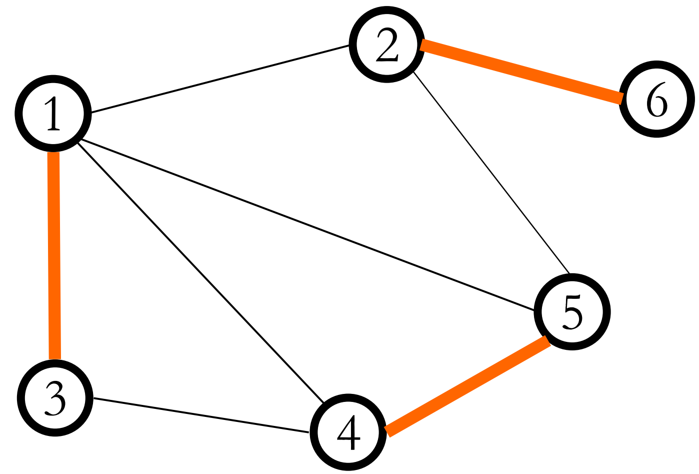
- $F = E$ è una copertura (banale) con archi.

Copertura con archi

$F \subseteq E$ è una copertura con archi se $\forall u \in V \ \exists \ uv \in F$.



Copertura con archi minimale



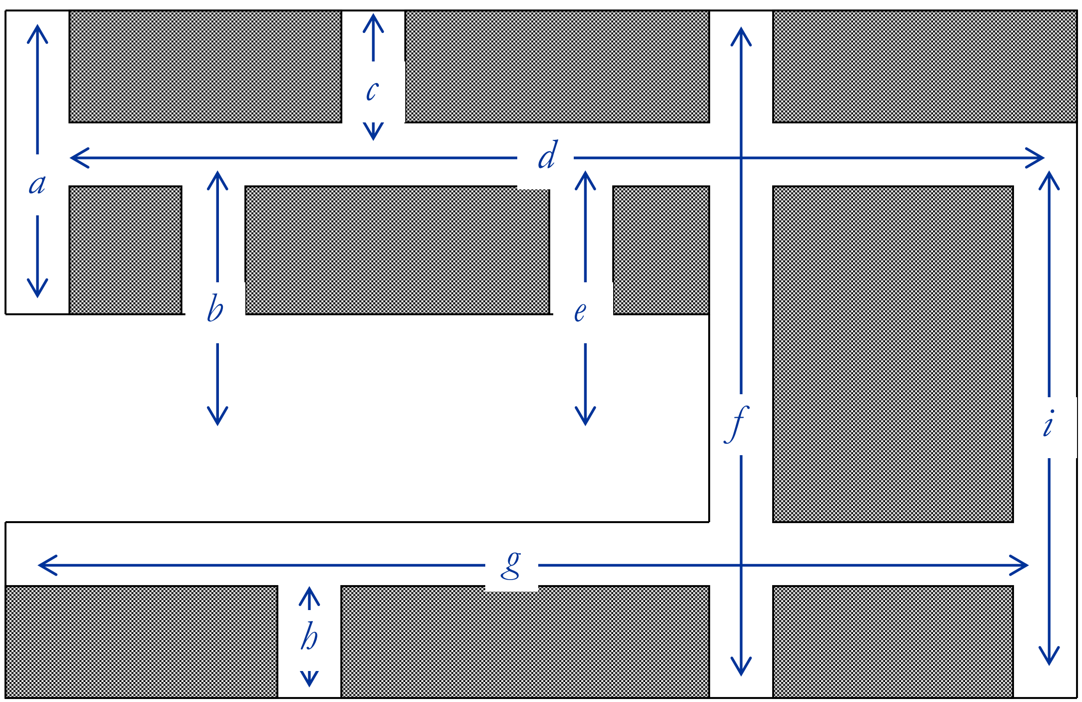
Copertura con archi minima

- una copertura con archi **non è** un abbinamento.
- Un albero ricoprente è una copertura con archi ma in generale non è vero il viceversa.

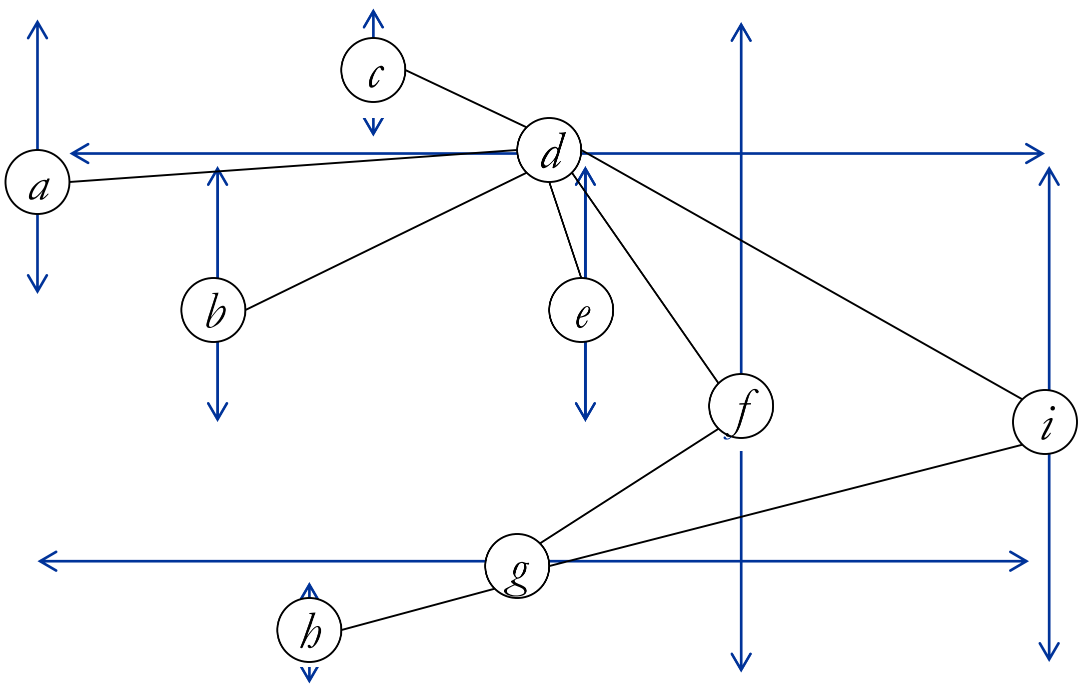
Esempi?

Applicazioni: il grande fratello

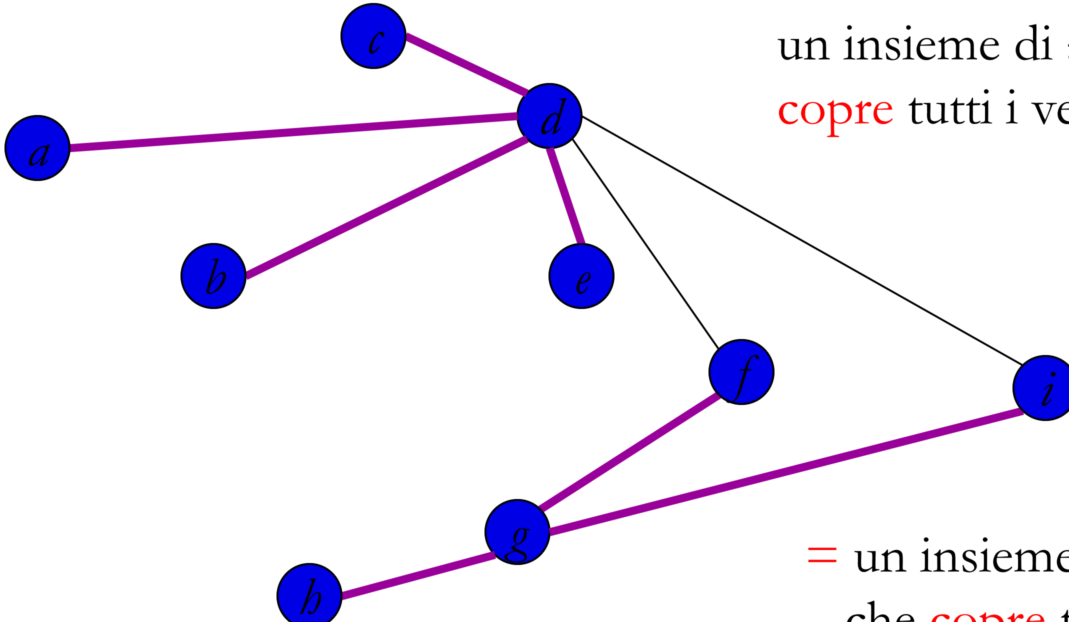
Si vuole dotare un museo di un sistema di telecamere per la sorveglianza in assenza di personale. Sapendo che una telecamera posta all'incrocio di due corridoi è in grado, con opportune rotazioni, di sorvegliarli entrambi, come disporre le telecamere?



Definiamo un grafo $G = (V, E)$ in cui i nodi rappresentano i corridoi e due nodi sono adiacenti se i corrispondenti corridoi si intersecano. Se non ci sono incroci tra più di due corridoi, ogni incrocio corrisponde ad un arco.



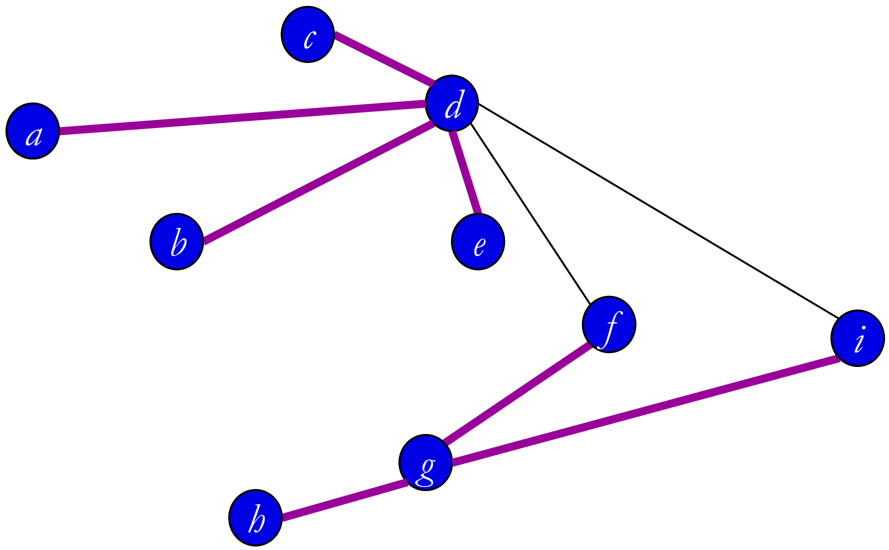
Quindi, possiamo interpretare la scelta di un arco *uv* come l'installazione di una telecamera all'incrocio dei corridoi *u* e *v*.

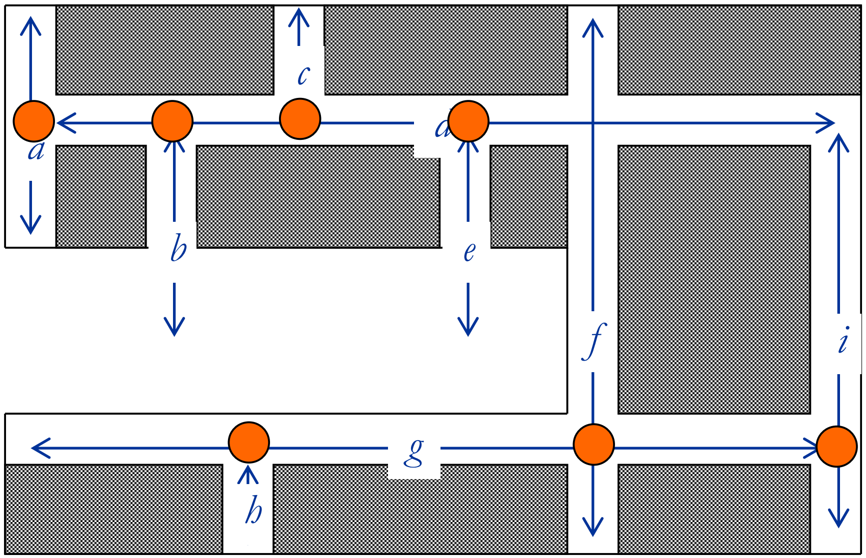


un insieme di archi che copre tutti i vertici

= un insieme di telecamere che copre tutti i corridoi

Soluzione:

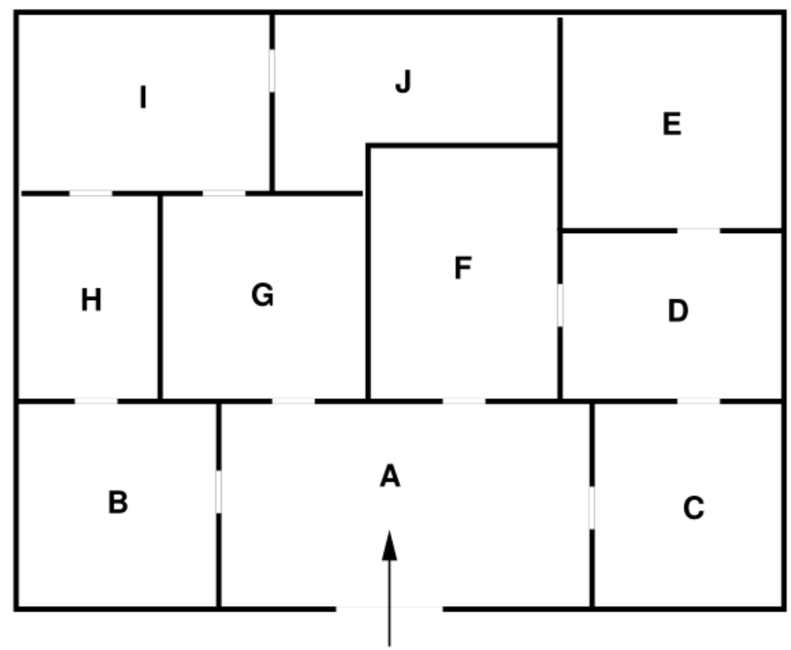




Esercizio: quali sono i limiti di una modellazione alternativa che prevede un nodo per ogni incrocio e un arco tra ogni coppia di incroci che condividono un corridoio?

Applicazioni: guardie e ladri …

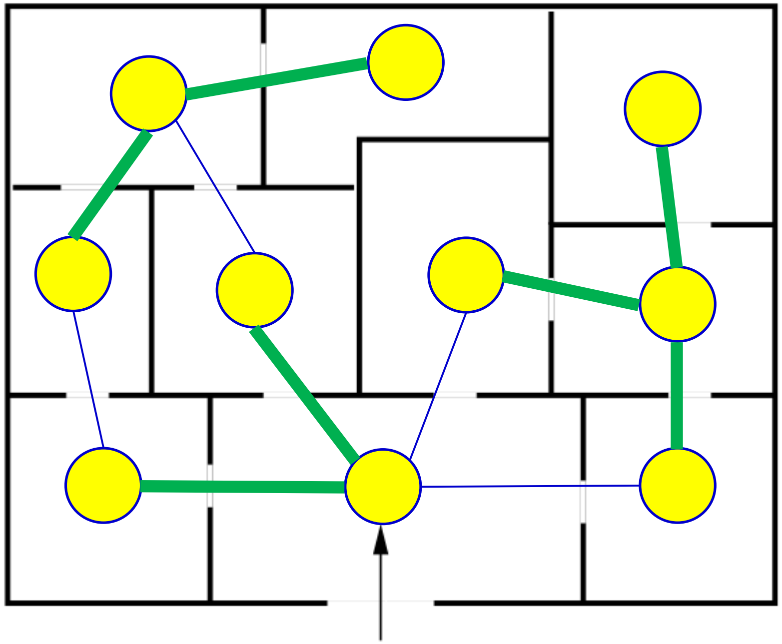
- Qual è i minimo numero di guardie necessarie per controllare tutte le stanze del museo? (una guardia alla porta controlla le due stanze che la porta collega)



sia $G = (V, E)$ un grafo con

- vertici ≡ stanze
- archi ≡ porte

Ogni porta collega esattamente 2 stanze e le stanze devono essere tutte controllate…



- Insiemi indipendenti e coperture
- $\boxed{\text{Relazioni tra le strutture}}$
- algoritmo Greedy

Disuguaglianze duali deboli

- $\alpha(G)$: cardinalità di un insieme stabile massimo
- $\mu(G)$: cardinalità di un abbinamento massimo
- $\rho(G)$: cardinalità di una copertura con archi minima
- $\tau(G)$: cardinalità di una copertura con nodi minima

[Teorema] Per ogni grafo G si ha:

$$\alpha(G) \leq \rho(G)$$
$$\mu(G) \leq \tau(G)$$

Disuguaglianze duali deboli: dimostrazione

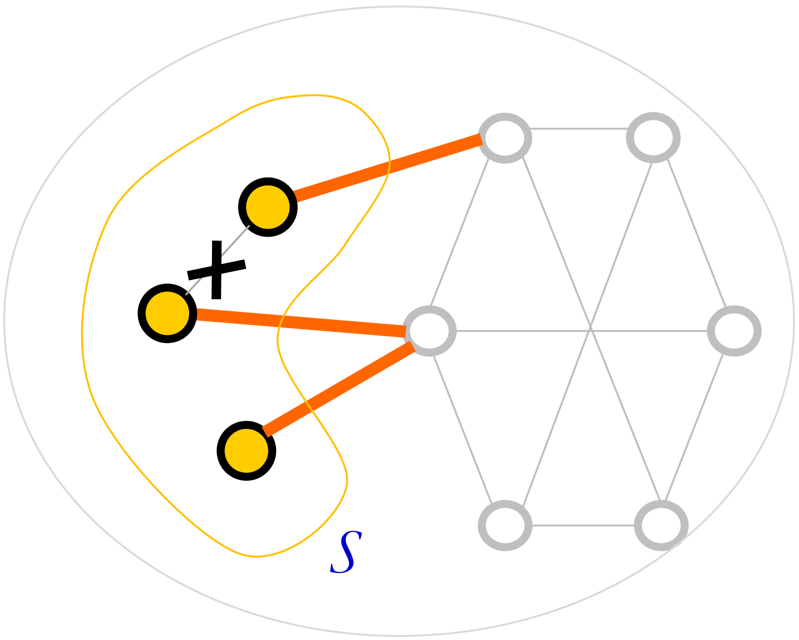
Per assurdo, siano $S \subseteq V$ un insieme stabile e $F \subseteq E$ una copertura con archi di G tali che $|\ S\ | > |\ F\ |$

- S è un insieme stabile quindi il sottografo di G indotto da S è vuoto.
- D'altra parte, F è una copertura con archi quindi ogni nodo di S deve essere estremo di almeno un arco di F
- Ciò significa che ci sono almeno tanti archi in F quanti nodi in S. assurdo. Quindi

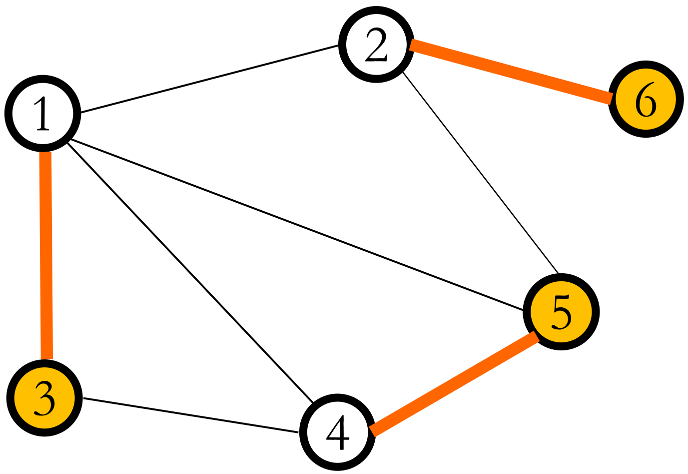
$$|\ S\ | \leq |\ F\ | \qquad \forall\ S, F$$

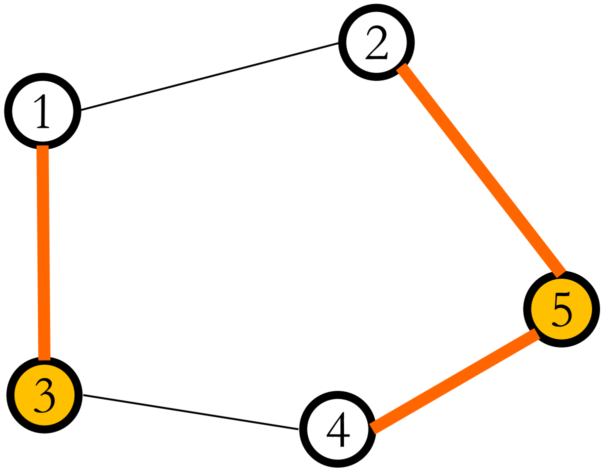
e in particolare $\alpha(G) \leq \rho(G)$

$G = (V, E)$



Esempio di $\alpha(G) \leq \rho(G)$





$S = \{3,5,6\}$

$F = \{13, 45, 26\}$

$|\ S\ | = |\ F\ |$

$S = \{3,5\}$

$F = \{13, 45, 25\}$

$|\ S\ | < |\ F\ |$

Disuguaglianze duali deboli: dimostrazione

La dimostrazione del secondo caso è analoga. Siano $M \subseteq E$ un abbinamento e $T \subseteq V$ una copertura con nodi di G tali che $|M| > |T|$

$G = (V, E)$

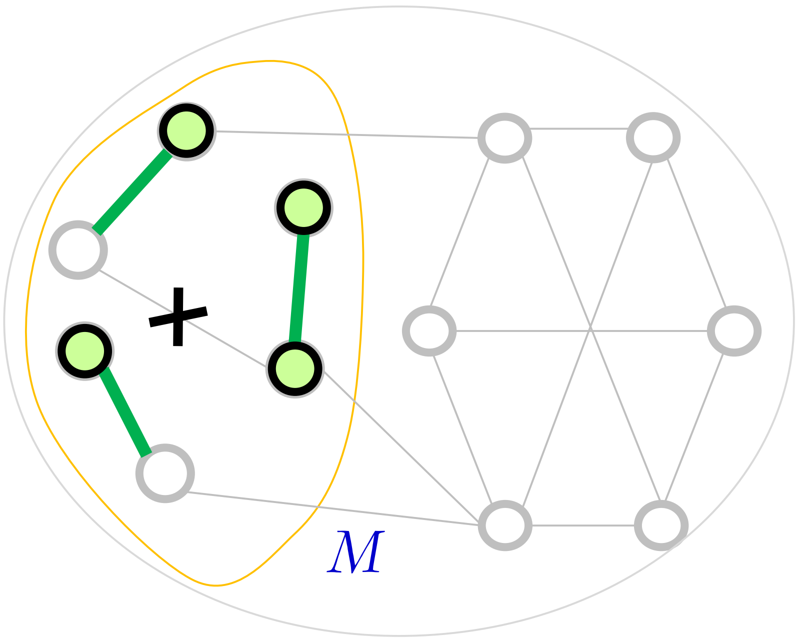
- M è un abbinamento quindi ogni nodi di G è estremo di al più un arco di M.

- D'altra parte, T è una copertura con nodi quindi ogni arco di M deve avere almeno un estremo in T

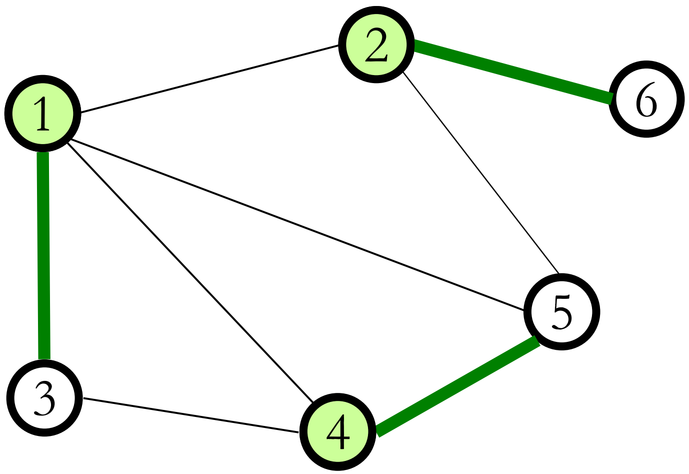
- Ciò significa che ci sono almeno tanti nodi in T quanti archi in M. assurdo. Quindi

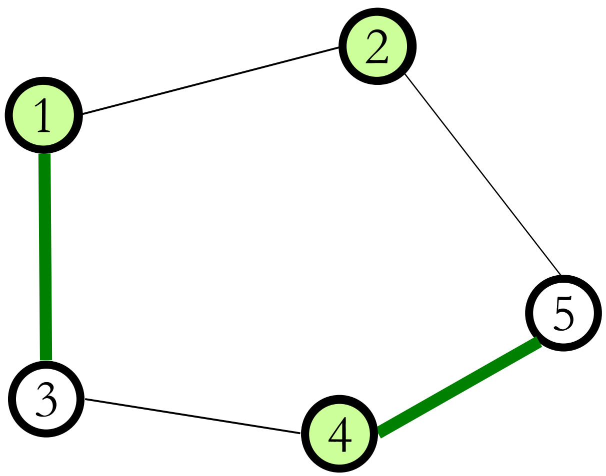
  $| M | \leq | T |$ $\forall M, T$

  e in particolare $\mu(G) \leq \tau(G)$



Esempio di $\mu(G) \leq \tau(G)$





$$M = \{13,45,26\}$$
$$T = \{1, 2, 4\}$$
$$|\ M\ | = |\ T\ |$$

$$M = \{13,45\}$$
$$T = \{1, 2, 4\}$$
$$|\ M\ | < |\ T\ |$$

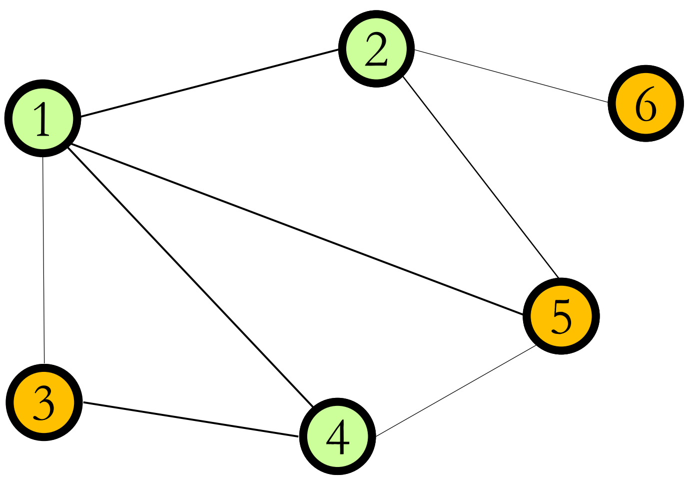
Teorema di Gallai

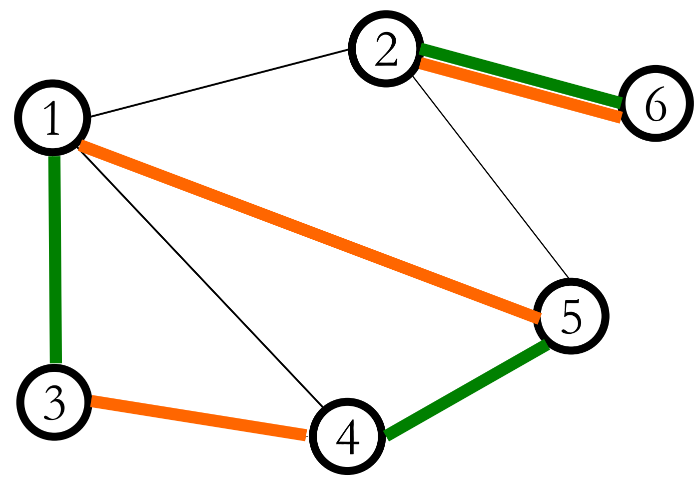
[Teorema] Per ogni grafo G con n nodi si ha:

$$\alpha(G) + \tau(G) = n$$

e se G non ha nodi isolati

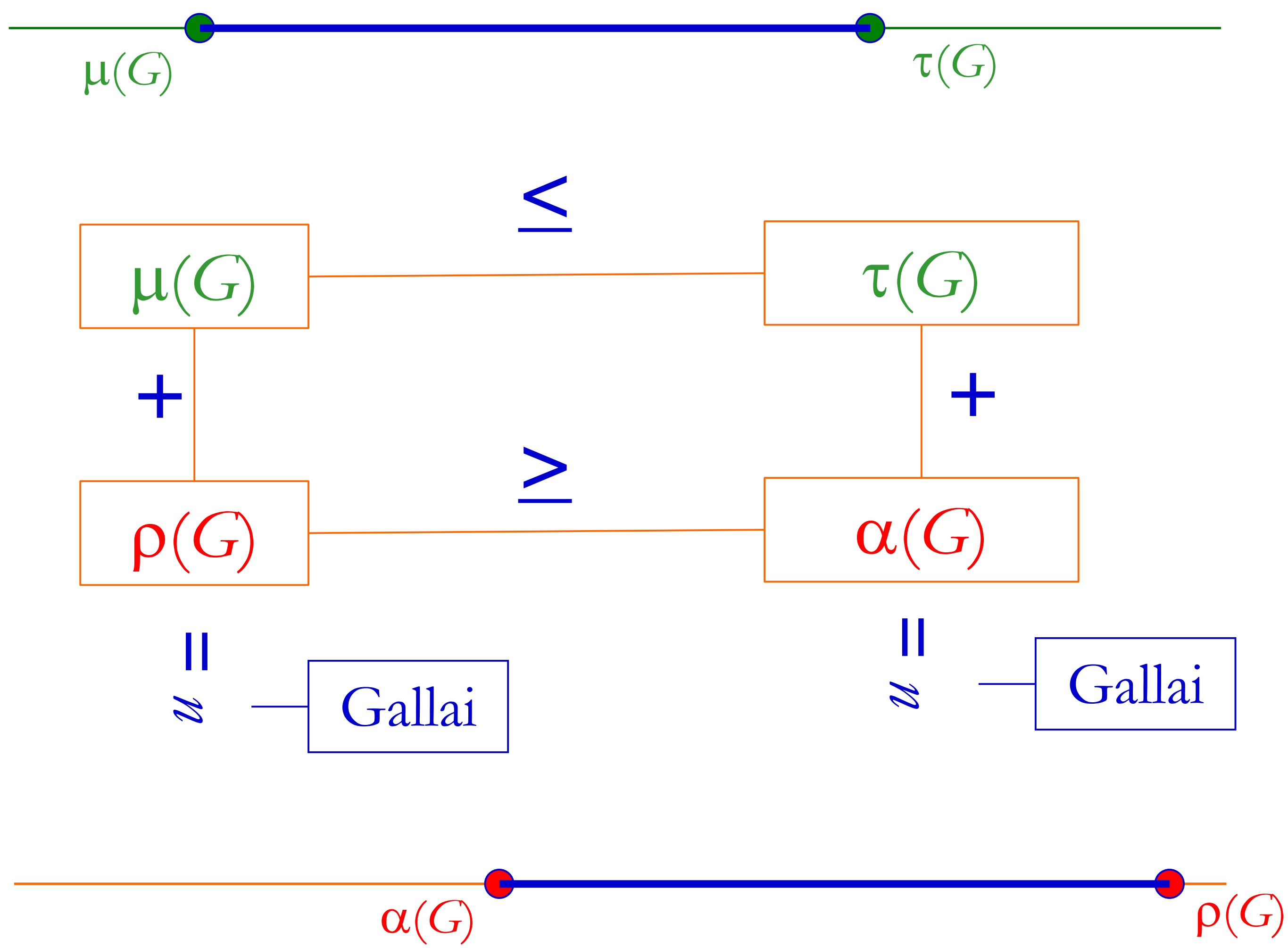
$$\mu(G) + \rho(G) = n$$





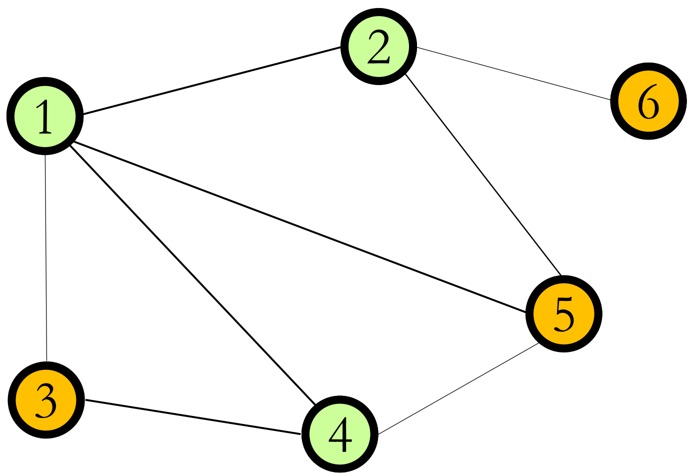
Schema riassuntivo

$$\rho(G) - \alpha(G) = \tau(G) - \mu(G)$$



Insieme stabile e copertura con nodi …

[Corollario] S è un insieme stabile di $G = (V, E)$ **se e solo se** $T = V \setminus S$ è una copertura con nodi di G



- S è un insieme stabile
- $\Leftrightarrow$ (def.) ogni arco è incidente al più su un nodo di S
- $\Leftrightarrow$ ogni arco è incidente su almeno un nodo in $V \setminus S$
- $\Leftrightarrow$ (def.) $V \setminus S$ è una copertura con nodi

- Insiemi indipendenti e coperture
- Relazioni tra le strutture
- algoritmo Greedy

Algoritmo *Greedy* (dell'ingordo): Idea

Sia U un insieme finito e discreto di elementi e $\wp$ un predicato che esprime una proprietà data.

Si costruisce una soluzione ammissibile (rispetto a $\wp$) considerando iterativamente gli elementi di U e scegliendo ad ogni passo l'elemento più «conveniente» in quel momento.

[osservazioni]

- l'algoritmo non rimettere mai in discussione le scelte fatte nelle precedenti iterazioni.
- La convenienza dell'elemento scelto è dettata dal problema.

Algoritmo *Greedy*

Sia U un insieme finito e discreto di elementi e $\wp$ un predicato che esprime una proprietà data.

Algoritmo Greedy

Inizializzazione: $X := \emptyset$

1) Scegli l'elemento $u \in U$ «più conveniente»

2) Se $X \cup \{u\}$ soddisfa $\wp$, allora $X := X \cup \{u\}$

3) Poni $U := U - \{u\}$; se $U \neq \emptyset$ torna a (1), altrimenti restituisci X e stop.

Algoritmo *Greedy*: vantaggi

- Problema dello zaino

	Profitto $f(u_i)$	*Peso* a_i
u_4	100	52
u_3	60	23
u_2	70	35
u_1	15	15

Capacità $b = 90$

2^n valutazioni

Enumerazione totale

Soluzione	*Ingombro*	*Profitto*	*Ottimo corrente*
$X_0 = \varnothing$	0	0	$\mathbf{X_0}$
$X_1 = \{u_1\}$	15	15	$\mathbf{X_1}$
$X_2 = \{u_2\}$	35	70	$\mathbf{X_2}$
$X_3 = \{u_3\}$	23	60	X_2
$X_4 = \{u_4\}$	52	100	$\mathbf{X_4}$
$X_5 = \{u_1, u_2\}$	50	85	X_4
$X_6 = \{u_1, u_3\}$	38	75	X_4
$X_7 = \{u_2, u_2\}$	58	130	$\mathbf{X_7}$
$X_8 = \{u_1, u_4\}$	67	115	X_7
$X_9 = \{u_2, u_4\}$	87	170	$\mathbf{X_9}$
$X_{10} = \{u_3, u_4\}$	75	160	X_9
$X_{11} = \{u_1, u_2, u_3\}$	72	145	X_9
$X_{12} = \{u_1, u_2, u_4\}$	102	(inammissibile)	X_9
$X_{13} = \{u_1, u_3, u_4\}$	90	175	$\mathbf{X_{13}}$
$X_{14} = \{u_2, u_3, u_4\}$	110	(inammissibile)	X_{13}
$X_{15} = \{u_1, u_2, u_3, u_4\}$	125	(inammissibile)	X_{13}

Algoritmo *Greedy*: vantaggi

- Problema dello zaino

	Profitto $f(u_i)$	*Peso* a_i
u_4	100	52
u_3	60	23
u_2	70	35
u_1	15	15

Capacità $b = 90$

Algoritmo *Greedy* (criterio: min ingombro)

Soluzione	*Ingombro*	*Profitto*	*sol. corrente*
$X_0 = \varnothing$	0	0	X_0
$X_1 = \{u_1\}$	15	15	X_1
$X_2 = \{u_1, u_3\}$	38	75	X_2
$X_3 = \{u_1, u_3, u_2\}$	72	145	X_4
$X_4 = \{u_1, u_2, u_3, u_4\}$	125	(inammissibile)	-

n valutazioni

Algoritmo *Greedy*: vantaggi

$\boxed{n << 2^n \text{ valutazioni}}$

Algoritmo *Greedy*: limiti

L'algoritmo Greedy determina **sempre** una soluzione ammissibile?

- L'algoritmo Greedy costruisce la soluzione finale aggiungendo elementi di U, cioè presupponendo che se X è una soluzione ammissibile, lo è anche qualsiasi sottoinsieme di X.

- Questa strategia non funziona sempre: per esempio se X è un *edge-cover*, non è detto che un sottoinsieme di X lo sia.

Un predicato $\wp$ si dice subclusivo se $\emptyset$ soddisfa $\wp$ e

$$\forall X \text{ che soddisfa } \wp, \quad Y \subseteq X \Rightarrow Y \text{ soddisfa } \wp$$

Algoritmo *Greedy*: limiti

- Gli insiemi indipendenti (ma anche le soluzioni del problema dello zaino) sono definiti da predicati subclusivi.

Se il predicato è subclusivo, la precedente codifica del greedy garantisce l'ammissibilità della soluzione calcolata.

- Il predicato relativo ad altri casi (edge-cover, trasversale, abbinamento perfetto,…) non è subclusivo. In alcuni di questi casi (edge-cover, trasversale) il predicato gode della proprietà complementare:

Un predicato $\wp$ si dice superclusivo se U soddisfa $\wp$ e

$$\forall X \text{ che soddisfa } \wp, \quad X \subseteq Y \Rightarrow Y \text{ soddisfa } \wp$$

Algoritmo *Greedy* per predicati superclusivi

Se il predicato è superclusivo si può tentare una variante del Greedy che ragioni per sottrazione a partire dalla soluzione X che coincide con U

Algoritmo Greedy

Inizializzazione: $X := U$

1) Scegli l'elemento $u \in U$ «più conveniente»

2) Se $X \setminus \{u\}$ soddisfa $\wp$, allora $X := X \setminus \{u\}$

3) Poni $U := U - \{u\}$; se $U \neq \emptyset$ torna a (1), altrimenti restituisci X e stop.

Algoritmo *Greedy*: limiti

L'algoritmo Greedy determina **sempre** una soluzione ottima?

Senza dubbio l'algoritmo Greedy fornisce soluzioni massimali per predicati subclusivi, e la variante del Greedy fornisce soluzioni minimali per predicati superclusivi.

Ma queste soluzioni sono anche massime (rispettivamente minime) ?

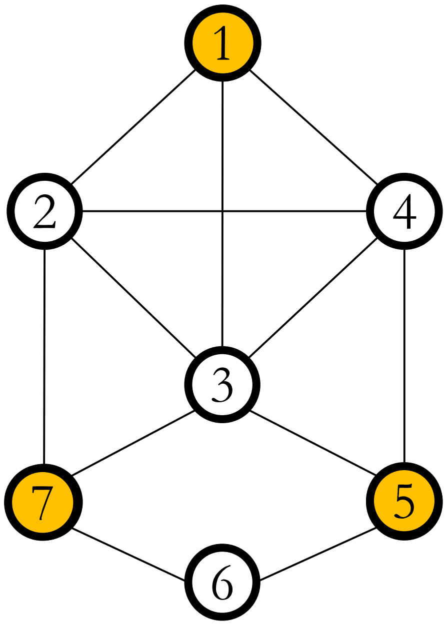
Algoritmo *Greedy* per insieme stabile

Algoritmo Greedy (per insieme stabile)

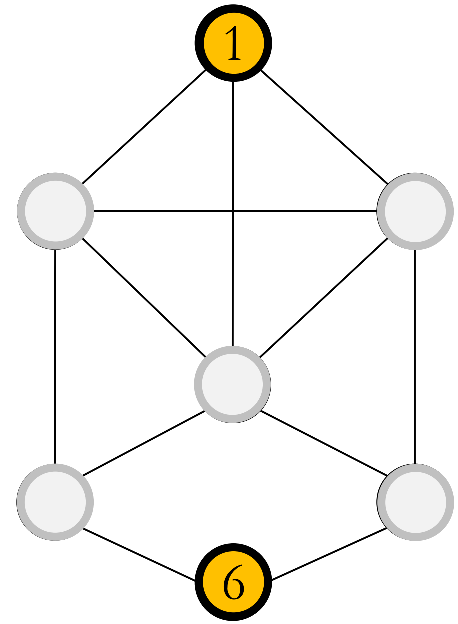
Inizializzazione: $X := \emptyset$ e $U := V$

1) Scegli il nodo $u \in U$ di **grado minimo**
2) Se $X \cup \{u\}$ è un insieme stabile, allora $X := X \cup \{u\}$
3) Poni $U := U - \{u\}$; se $U \neq \emptyset$ torna a (1), altrimenti restituisci X e stop.

Quando l'algoritmo *Greedy* fallisce



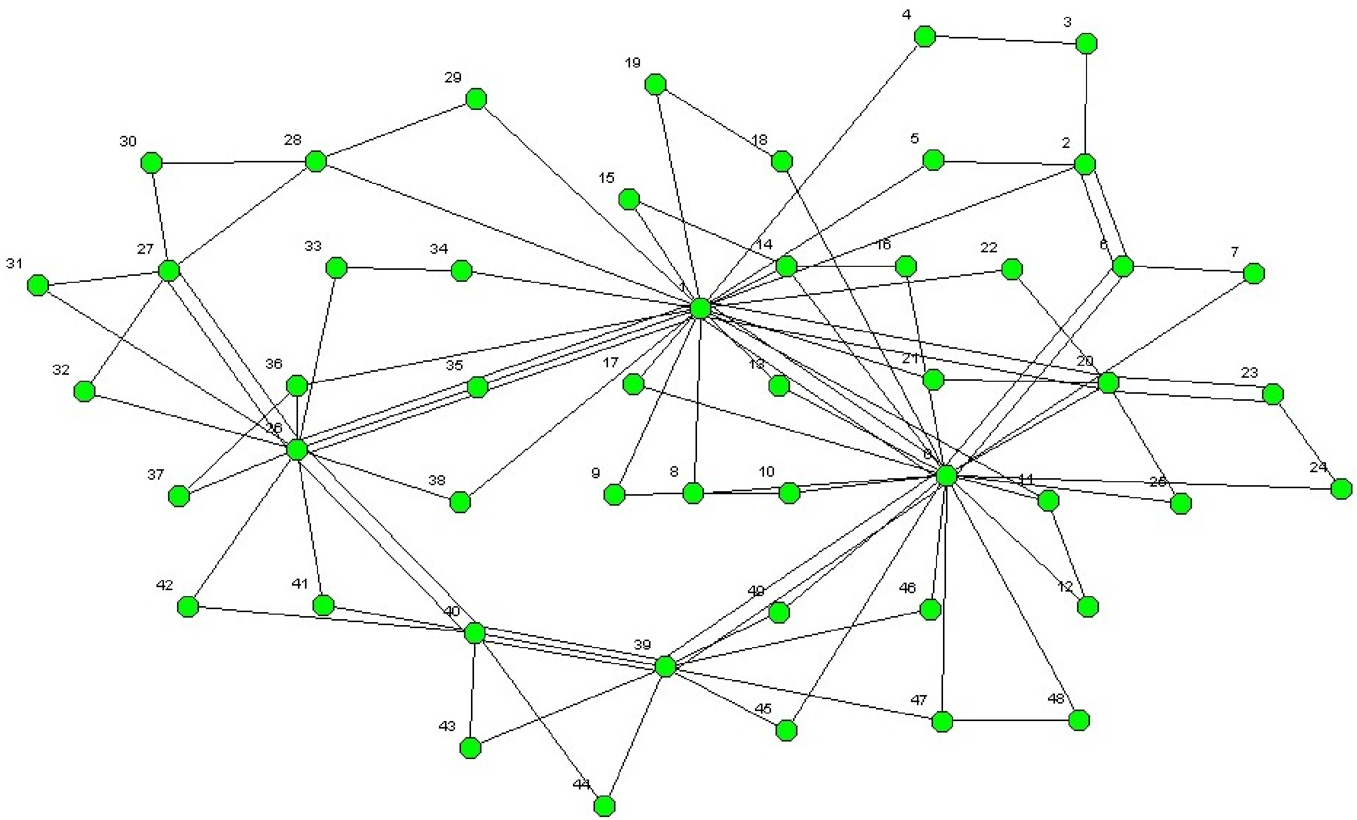
massimo insieme stabile



Soluzione del greedy

Il greedy non è un algoritmo esatto per il problema di massimo insieme stabile

Esercizi



- Determinare (con algoritmi *greedy*) un insieme stabile, un abbinamento, una copertura con archi e una copertura con nodi del grafo in figura.

Algoritmo *Greedy*: limiti

Esaminando la soluzione del greedy, notiamo che l'algoritmo fallisce perché esistono insiemi massimali che non sono massimi… e questo non è un caso.

[Teorema] Se la proprietà $\wp$ definisce massimali di cardinalità diversa, allora è sempre possibile costruire una funzione peso $c: U \to \mathbb{R}$ tale che l'algoritmo greedy produca una soluzione massimale ma non massima

Algoritmo *Greedy*: limiti

Ci sono casi in cui l'algoritmo Greedy fornisce soluzioni ottime?

Sì… quando il problema è un matroide

Matroidi

Sia U un insieme finito e discreto di elementi, $\wp$ un predicato che esprime una proprietà data, e $\mathfrak{I}$ la famiglia di sottoinsiemi di U che soddisfano $\wp$.

[Definizione] La coppia $(U, \mathfrak{I})$ è un matroide se

- $\wp$ è un predicato subclusivo
- vale la proprietà di scambio:

> per ogni $X, Y \in \mathfrak{I}$ con $|X| < |Y|$, <u>esiste sempre</u> un $y \in Y - X$ tale che $X \cup \{y\} \in \mathfrak{I}$.

Matroidi e algoritmo Greedy

[Teorema] (Rado)

Qualunque sia la funzione peso $c: U \to \mathbb{R}$, l'algoritmo greedy determina un insieme (pesato) massimo che soddisfa $\wp$

se e solo se

$(U, \mathfrak{I})$ è un matroide

Un esempio «concreto» di matroide?

Letture ricreative e contenuti multimediali

1. P. M. Higgins,
   La matematica dei social network. Una introduzione alla teoria dei grafi
   Dedalo, 2011

2. Brian Eno
   Thursday Afternoon
   1985