(1) GRAFI

- **HANDSHAKING LEMMA**: $\sum_{v \in V} \deg(v) = 2 \cdot |E(G)|$, per ogni grafo $G(V,E)$ non orientato

$\Rightarrow$ * # vertici di grado dispari è PARI !

- **GRAFO COMPLEMENTO**: $|E| + |\bar{E}| = \frac{|V|(|V|-1)}{2} = \binom{|V|}{2} = \binom{n}{2} = \frac{n(n-1)}{2}$

- **CIRCUITO EULERIANO**: Circuito (trail chiuso, più volte sullo stesso nodo ma non sullo stesso arco) tale che visiti OGNI SPIGOLO del grafo ESATTAMENTE 1 volta!

Sia $G = (V,E)$ connesso. Sono equivalenti le seguenti affermazioni:
(1) G ammette un CIRCUITO EULERIANO;
(2) $\forall v \in V(G)$: $\deg(v)$ è PARI;
(3) $E(G)$ può essere PARTIZIONATO in CIRCUITI $C_1, C_2, \dots, C_e$.

ALGORITMO DI HIERHOLZER:
1. Partizionare il grafo in circuiti che tornano al nodo di partenza;
2. Concatenare i circuiti, fino a quando non ne rimane solo 1;
3. Tale circuito sarà un CIRCUITO EULERIANO.

* Se G ha AL PIU' 2 vertici di GRADO DISPARI (0 oppure 2) $\Rightarrow$ G ammette TRAIL EULERIANO

$\rightarrow$ aggiungere uno spigolo ai vertici di grado dispari!

- **CONNESSIONE, ACICLICITÀ**:

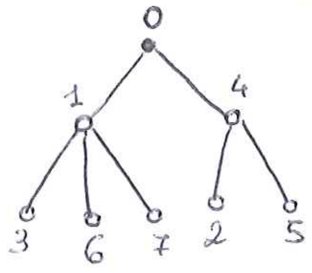
$$G(V,E): \begin{cases} \text{CONNESSO} \rightarrow |E(G)| \geq |V(G)| - 1 \\ \text{ACICLICO} \rightarrow |E(G)| \leq |V(G)| - 1 \\ \text{ALBERO} \rightarrow |E(G)| = |V(G)| - 1 \end{cases}$$

(connesso e aciclico)

- **LEMMA FOGLIE ALBERO**: Un qualunque albero ha ALMENO 2 FOGLIE, cioè vertici di grado 1.

- **GROWING TREE PROCEDURE**: Partendo da un vertice, aggiungendo una FOGLIA ad ogni passo, si ottiene sempre un'albero.

- **PRÜFER CODE**: Array di $(n-2)$ elementi che rappresentano un ALBERO di n vertici; gli $n-2$ dell'array sono i <u>PADRI</u>, dove l'$n-1$ esimo è lo 0, sottointeso. Si parte dalla FOGLIA PIÙ BASSA (0 escluso), usando la GTP al contrario, in rimozione.



$a = [2\ 3\ 5\ 4\ 6\ 7\ 1]$

$b = [4\ 1\ 4\ 0\ 1\ 1]$ ← PRÜFER CODE

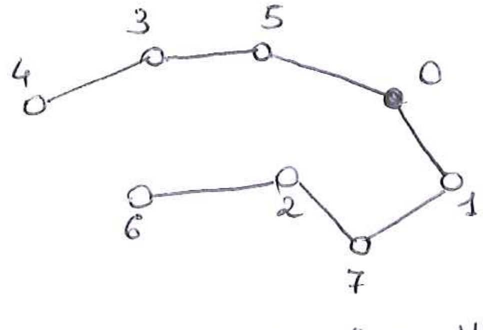
Da Prüfer Code → ALBERO:

$b = [3\ 5\ 0\ 2\ 7\ 1] \Rightarrow n-2 = 6 \Rightarrow n = 8$

In b ci sono i PADRI, quindi le foglie sono $\{0, \dots, n-1\} \setminus \{b\}$.

In questo caso: $\{0, 1, \dots, 7\} \setminus \{0, 1, 2, 3, 5, 7\} = \{4, 6\}$ ← FOGLIE

Prendere le foglie più basse volta per volta e ricordarsi di far diventare foglie i vertici del vettore b che man mano si leggono verso destra.



* MINIMO NUMERO DI BIT NECESSARI: $(n-2)\lceil \log_2 n \rceil$

* I diversi alberi con insieme dei vertici $V(T) = \{0, 1, \dots, n-1\}$ sono:

$$2^{(n-2)\lceil \log_2 n \rceil} = \boxed{n^{(n-2)}}$$

- **BAMBINI E MONETE**:
  - se almeno 1 moneta: $\boxed{C(n-1, k-1)}$
  - se 0 monete è ammesso: $\boxed{C(n+k-1, k-1)}$

(2) CONTEGGIO

- **REGOLA DEL PRODOTTO:** Se ho una richiesta di 2 compiti (ENTRAMBI) che possono essere eseguiti in n_1 ed n_2 modi diversi, allora il processo totale può essere fatto in

$$n_1 \cdot n_2 \quad \text{MODI DIVERSI}$$

- **REGOLA DEL PRODOTTO:** Se ho una SCELTA ALTERNATIVA $\rightarrow$ $n_1 + n_2$ MODI DIVERSI

  * Se la scelta è ALTERNATIVA, ma NON DISGIUNTIVA tra 2 insiemi A e B: $|A| + |B| - |A \cap B|$

- **PIGEONHOLE PRINCIPLE:** Se n oggetti devono essere collocati in k scatole, con $k < n$, ci sarà almeno 1 scatola che conterrà $\left\lceil \frac{n}{k} \right\rceil$ oggetti.

- **PERMUTAZIONE:** Disposizioni ORDINATE di un insieme di n oggetti è $n!$

  * R-PERMUTAZIONE: disposizioni ordinate di r elementi di un insieme che ha n elementi: $\frac{n!}{(n-r)!} = P(n,r)$

- **R-COMBINAZIONE:** Sottoinsieme NON ORDINATO di r elementi, a partire da un insieme di n elementi: $\boxed{C(n,r) = \binom{n}{r} = \frac{n!}{(n-r)! \cdot r!}}$

PROPRIETÀ:

1) $C(n,r) = C(n,n-r)$
2) $|E(K_n)| = \frac{n(n-1)}{2} = \binom{n}{2} = C(n,2)$
3) Identità di Pascal: $\left[C(n+1,r) = C(n,r) + C(n,r-1) \right]$
4) Binomio di Newton: $\sum_{k=0}^{n} C(n,k) = \sum_{k=0}^{n} \binom{n}{k} = 2^n$

$$\boxed{(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k}$$

(3) COLORAZIONE

- **COLORAZIONE AMMISSIBILE:** $f : V(G) \longmapsto \{1, 2, \dots, m\}$ tale che:

  $\forall \{u,v\} \in E(G) : f(u) \neq f(v)$ → vertici adiacenti sono colorati diversamente

- **NUMERO CROMATICO:** Anche detto "INDICE DI COLORAZIONE", è il più piccolo numero K tale che G sia K-colorabile : $\chi(G) = K$.

- **CLIQUE NUMBER:** È la dimensione della più grande CLIQUE di G := $\omega(G)$

  Rappresenta un lower bound per $\chi(G)$: $\boxed{\chi(G) \geq \omega(G)}$

- **TEOREMA BICOLORABILITÀ:** [Un grafo G(V,E) è BIPARTITO (cioè $\chi(G) = 2$) $\Longleftrightarrow$ G NON ammette CICLI DISPARI]

  ALGORITMO BIPARTIZIONE:

  1. Scegliere un vertice $v \in V(G)$ e designarlo come "radice";
  2. Tramite BFS, calcolare per ogni nodo la distanza dalla radice;
  3. Nodi con la stessa distanza formano una classe; se non ci sono archi tra nodi di una stessa classe, si ha una BIPARTIZIONE, altrimenti si ha un CICLO DISPARI, che è CERTIFICATO che una bipartizione non esiste!

- **GRAFO INTERVALLO:**
  - NON ammettono cicli senza corde di lunghezza > 3.
  - $\chi(G) = \omega(G)$
  - COLORAZIONE OTTIMA con l'ALGORITMO GREEDY se si ordinano i vertici per estremo sinistro di intervallo crescente.

* In un GRAFO BIPARTITO CONNESSO, esiste 1 sola COLORAZIONE!

- **BOUNDS PER $\chi(G)$:** $\boxed{\omega(G) \leq \chi(G) \leq \Delta(G) + 1}$

  dove $\Delta(G) := \max \{\deg_G(v)\}$.

- **EDGE-COLORING:** È una funzione $f: E(G) \longmapsto \{1,2,\dots,k\}$ tale che:

  $\forall e_1, e_2 \in E(G)$, se e_1 ed e_2 sono incidenti $\Rightarrow f(e_1) \neq f(e_2)$

- **EDGE CHROMATIC NUMBER:** $\chi'(G)$; numero più basso K affinché sia vero che G è edge-K-colorabile.

- **TEOREMA DI VIZING:** $\boxed{\Delta(G) \le \chi'(G) \le \Delta(G)+1}$

Non è noto alcun algoritmo polinomiale per decidere se $\chi'(G)=\Delta(G)$ oppure $\chi'(G)=\Delta(G)+1$.

C'è però un caso particolare:

* Per un grafo COMPLETO K_n:

$$\chi'(K_n) = \begin{cases} n-1, & \text{se } n \text{ PARI} \\ n, & \text{se } n \text{ DISPARI} \end{cases}$$

(4) PROBLEMI DI FLUSSO

Abbiamo sempre un GRAFO <u>ORIENTATO</u> $D(N,A)$, due nodi $s,t \in N(D)$ detti "sorgente" e "destinazione" e una FUNZIONE CAPACITÀ $u : A(D) \longmapsto \mathbb{Z}_+$.

- **VETTORE DI FLUSSO**: $f(u,v) :=$ numero di cammini $\{P_1, \dots, P_k\}$ della soluzione che utilizzano l'arco (u,v).

  Il <u>VALORE DEL FLUSSO</u> è la somma dei flussi uscenti da S meno la somma dei flussi entranti in S:

$$\mathrm{Val}(f) := \sum_{(s,v) \in A} f(s,v) - \sum_{(v,s) \in A} f(v,s)$$

- **FLUSSO S-t AMMISSIBILE**: $f : A(D) \longmapsto \mathbb{Z}_+$ è s-t ammissibile se e solo se:

$$\begin{cases} (1) \quad f(u,v) \leq u(u,v), & \forall (u,v) \in A(D); \\ (2) \quad \displaystyle\sum_{u:(v,u) \in A} f(v,u) = \sum_{u:(u,v) \in A} f(u,v), & \forall v \in N(D),\ v \neq s,t \end{cases}$$

- **FLUSSO ACICLICO**: Un vettore di flusso f è ACICLICO se il grafo $D(N, A(f))$ è ACICLICO, dove: $A(f) = \{(u,v) \in A : f(u,v) > 0\}$

  Cioè il grafo di partenza, mantenendo solo gli archi su cui transita flusso.

  * Si può sempre ottenere un FLUSSO ACICLICO, rimuovendo una quantità di flusso ε da OGNI ARCO del CICLO, fin quando un arco non diventi a flusso nullo, ma mai negativo.

- **TAGLIO S-t**: È una PARTIZIONE dei vertici $N(D)$ in 2 classi V_1 e V_2 tali che: $s \in V_1$ e $t \in V_2$; si indica con $[V_1, V_2]$.

- **ARCHI CHE ATTRAVERSANO IL TAGLIO**: Sono di 2 tipi:
  - <u>CONCORDI</u>: $(u,v) \in A$: $u \in V_1$, $v \in V_2$ $\rightarrow$ da V_1 a V_2
  - <u>DISCORDI</u>: $(u,v) \in A$: $u \in V_2$, $v \in V_1$ $\rightarrow$ da V_2 a V_1

- FLUSSO NETTO ATTRAVERSO $[X, \overline{X}]$:

$$\left[\sum_{\substack{(u,v)\in A:\\ u\in X\\ v\in \overline{X}}} f(u,v) - \sum_{\substack{(u,v)\in A\\ u\in \overline{X}\\ v\in X}} f(u,v) = val(f)\right] \rightarrow$$

Flusso da V_1 a V_2 $-$ flusso da V_2 a V_1 $=$ $val(f)$

- UPPER BOUND al flusso netto:

$$\sum_{\substack{(u,v)\in A\\ u\in X\\ v\in \overline{X}}} f(u,v) - \sum_{\substack{(u,v)\in A\\ u\in \overline{X}\\ v\in X}} f(u,v) \leq \sum_{\substack{(u,v)\in A:\\ u\in X\\ v\in \overline{X}}} u(u,v)$$

CAPACITÀ DEL TAGLIO: Somma delle CAPACITÀ dei SOLI ARCHI CONCORDI attraverso il taglio $\rightarrow$ dipende dallo specifico taglio!

- ALGORITMO DI FORD-FULKERSON DEI CAMMINI AUMENTANTI:

Abbiamo 2 tipologie di cammini aumentanti:

- ORIENTATI da s a t con archi non saturi (semplici);
- NON NECESSARIAMENTE ORIENTATI da s a t tali che:
  - arco concorde alla percorrenza da s a t NON SATURO;
  - arco discorde al verso di percorrenza da s a t NON VUOTO.

1. Partendo da s, tramite DFS, trovare cammini aumentanti del tipo visti sopra;
2. Aggiornare i flussi ed iterare;
3. Quando da s non è più possibile raggiungere t, s unito ad i nodi da esso raggiungibili, forma una partizione per N(D) che individua il TAGLIO DI CAPACITÀ MINIMA.

$$\boxed{\text{MASSIMO FLUSSO} \equiv \text{CAPACITÀ MINIMO TAGLIO}}$$

- RETE RESIDUA RISPETTO AL FLUSSO CORRENTE:

CAMMINI del 1° TIPO $\rightarrow$ Possiamo la rete togliendo gli ARCHI SATURI

CAMMINI del 2° TIPO $\rightarrow$ Archi SATURI : verso di percorrenza invertito
Archi Vuoti : verso di percorrenza originale
Archi non vuoti e non saturi: percorribili in 2 direzioni

• Generalizzazione del Problema di Flusso:

Se abbiamo più nodi sorgente e/o più nodi destinazione:

- aggiungiamo un nodo sorgente generale con archi verso le altri sorgenti;
- analogamente un nodo destinazione generico, con archi dalle destinazioni alla destinazione generica;
- la capacità degli archi aggiunti è pari a ∞.

• Taglio multi s - multi t:

Partizione di N in 2 classi X e $\overline{X}$ con X che contiene TUTTE le sorgenti e $\overline{X}$ che contiene TUTTE le destinazioni.

[Valore flusso multisorgente $\equiv$ Valore flusso s-t corrispondente]

• Matching in grafi bipartiti:

Insieme di ARCHI a coppie NON INCIDENTI.

Dato un GRAFO BIPARTITO $G(X \cup Y, E)$:

se $\exists\, Q \subseteq X$ tale che $|N(Q)| < |Q|$ $\Rightarrow \nexists$ matching $|X|$-completo!

con $N(Q) = \{v \in Y : \exists (u,v) \text{ con } u \in Q\}$

Cioè, se esiste un insieme di n nodi che ha archi verso m nodi uguali, con $m < n$.

• Teorema del Matrimonio:

[Condizione NECESSARIA e SUFFICIENTE affinché un grafo bipartito $G(X \cup Y, E)$ ammette un MATCHING $|X|$-COMPLETO è che:

$$\forall Q \subseteq X \text{ vale che: } |N(Q)| \geq |Q|$$
]

• Come trovare un Matching di Valore Massimo:

Trasformo il problema in un problema di flusso, aggiungendo 2 nodi s e t:

- s collegato ai nodi di X con archi di capacità 1;
- t collegato dai nodi di Y a t con archi di capacità 1;
- orientamento degli archi da s verso t e quelli esistenti hanno capacità ∞

(5) PROGRAMMAZIONE LINEARE

- **PL**: Un problema di programmazione lineare (PL) è un problema di ottimizzazione di tipo max/min in cui:
  1. la FUNZIONE OBIETTIVO è LINEARE;
  2. la regione ammissibile (vincoli) è definita da un insieme FINITO di disuguaglianze LINEARI di tipo $\geq$, $\leq$ o $=$.

Un problema $\mathcal{P}$ di PL può essere scritto nella forma:

$$\boxed{\begin{array}{l} \max\ c^t x \\ \quad Ax \leq b \end{array}}\ , \text{ con: } \left.\begin{array}{l} c \in \mathbb{Z}^n \\ A \in \mathbb{Z}^{m \times n} \\ b \in \mathbb{Z}^m \end{array}\right\} \text{ DATI DEL PROBLEMA}$$

$x \in \mathbb{R}^n \longrightarrow$ VETTORE DELLE VARIABILI

Infatti, valgono le seguenti equivalenze:

- $\max c^t x = -\min (-c)^t x$
- $a^t x \geq a_0 \iff (-a)^t x \leq -a_0$
- $a^t x = a_0 \iff \begin{cases} a^t x \leq a_0 \\ (-a)^t x \leq -a_0 \end{cases}$

SOLUZIONE AMMISSIBILE: $x \in \mathbb{R}^n$ che soddisfa i vincoli del sistema $Ax \leq b$.

SOLUZIONE OTTIMA: $x^* \in \mathbb{R}^n$ è OTTIMA se è ammissibile e se $\forall x \in \mathbb{R}^n$, soluzione ammissibile, vale:

$$\left[c^t x^* = \sum_{j=1}^{n} c_j x_j^* \geq \sum_{j=1}^{n} c_j x_j = c^t x \right]$$

- **POLIEDRO (RAZIONALE)**:

$P \subseteq \mathbb{R}^n$ è un POLIEDRO (razionale) se esistono $A \in \mathbb{Z}^{m \times n}$ e $b \in \mathbb{Z}^m$ tali che $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. Definisce l'insieme delle soluzioni ammissibili di un problema $\mathcal{P}$ di PL.

- **CONVESSITÀ**:

Ogni POLIEDRO è un insieme CONVESSO!

* Un insieme $\underset{\mathbb{R}^n}{\underset{\cap}{Q}}$ è convesso se e solo se: $\forall x, y \in Q : \left(\frac{1}{2}x + \frac{1}{2}y\right) \in Q$

- **Teorema Fondamentale della Programmazione Lineare:**

Sia $\mathcal{P}$ un problema di PL, il cui insieme delle soluzioni ammissibili è definito dal poliedro P. Se $\mathcal{P}$ ammette soluzione ottima, allora esiste una soluzione OTTIMA su un <u>VERTICE</u> del poliedro P.

- **PLI:**

È un problema della forma:

$$\boxed{\begin{aligned} &\max\ c^t x \\ &\quad Ax \le b \\ &\quad x \in \mathbb{Z}^m \end{aligned}}$$

$x \in \mathbb{Z}^m$ → Vincoli di Interezza

↓

<u>NON</u> sono esprimibili tramite VINCOLI LINEARI !

Sia $\mathcal{P}_I$ un PLI. Il problema lineare $\mathcal{P}_L$ che otteniamo da $\mathcal{P}_I$ rimuovendo i vincoli di interezza è detto "<u>RILASSAMENTO LINEARE di $\mathcal{P}_I$</u>".

Il valore della SOLUZIONE OTTIMA di $\mathcal{P}_I$ è sempre non migliore di quello della soluzione ottima di $\mathcal{P}_L$. Cioè:

$$\begin{aligned} &\max\ c^t x \\ &Ax \le b \\ &x \in \mathbb{Z}^m \end{aligned} \quad \le \quad \begin{aligned} &\max\ c^t x \\ &Ax \le b \end{aligned}$$

- **Set Covering e Set Packing:**

Si consideri un insieme di base B ed una famiglia di sottoinsiemi $\mathcal{R} \subseteq \{R \subseteq B\}$.

- **SET COVERING:** Un set covering di $\mathcal{R}$ consiste in un sottoinsieme $F \subseteq B$ tale che: $|F \cap R| \ge 1$, $\forall R \in \mathcal{R}$.

$$\boxed{\begin{aligned} &Ax \ge \mathbb{1} \\ &\quad x \in \{0,1\}^{|B|} \\ &\quad A \in \{0,1\}^{|R| \times |B|} \end{aligned}}$$

- **SET PACKING:** Sottoinsieme $F \subseteq B$ tale che: $|F \cap R| \le 1$, $\forall R \in \mathcal{R}$

$$\boxed{\begin{aligned} &Ax \le \mathbb{1} \\ &\quad x \in \{0,1\}^{|B|} \\ &\quad A \in \{0,1\}^{|R| \times |B|} \end{aligned}}$$

- **PROBLEMA DEL MASSIMO MATCHING:**

Dato un grafo $G=(V,E)$ e un vettore di peso $c \in \mathbb{Z}^{|E|}$, vogliamo trovare un matching $M \subseteq E(G)$ che massimizzi la somma dei pesi.

Se $c_e = 1 \;\forall e \in E(G)$, cioè $c^t = \mathbb{1} \Rightarrow$ PROBLEMA DEL MATCHING DI CARDINALITÀ MASSIMA

$$\left[\begin{array}{ll} \max & c^t x \\ & \sum_{e \in \delta(v)} x_e \leq 1 \quad , \forall v \in V(G) \\ & x \in \{0,1\}^{|E|} \end{array}\right]$$

* La funzione obiettivo può essere esplicitata come: $\max \sum_{e \in E(G)} c_e \cdot x_e$;

* I vincoli: $\sum_{e \in \delta(v)} x_e \leq 1$, $\forall v \in V(G)$ possono essere scritti come: $Hx \leq \mathbb{1}$,

con $H \in \{0,1\}^{|V| \times |E|}$ <u>MATRICE DI INCIDENZA</u> DI G $\rightarrow$ È un SET PACKING !

- **PROBLEMA DEL MASSIMO INSIEME STABILE:**

Dato un grafo $G(V,E)$, con $|V| = n$, un insieme stabile è un sottoinsieme $S \subseteq V$ di vertici NON ADIACENTI tra loro. Sia $c \in \mathbb{Z}^{|V|}$ un vettore dei pesi:

$$\left[\begin{array}{ll} \max & c^t y \\ & y_u + y_v \leq 1 \quad , \forall \{u,v\} \in E \\ & y \in \{0,1\}^{|V|} \end{array}\right]$$

* In questo caso si può utilizzare la MATRICE DI INCIDENZA TRASPOSTA:

$H^t y \leq \mathbb{1}$ $\longrightarrow$ SET PACKING

- **EDGE COVER e VERTEX COVER:**

Si ottengono rispettivamente dai problemi di massimo matching e di max stable set, ma MINIMIZZANDO:

$\min\ c^t x$	$\min\ c^t y$
$Hx \geq \mathbb{1}$	$H^t y \geq \mathbb{1}$
$x \in \{0,1\}^{\|E\|}$	$y \in \{0,1\}^{\|V\|}$

$\longrightarrow$ SET COVERING !

• DISGIUNZIONE:

Utile nei problemi di SCHEDULING per modellare attività che utilizzano risorse condivise.

Introduciamo: $x_{ij} = \begin{cases} 1 & \text{, se attività } i \text{ precede } j \\ 0 & \text{, altrimenti} \end{cases}$

Siano t_i il tempo di inizio processamento e p_i il tempo di processamento, allora:

$$\begin{bmatrix} t_j \geq t_i + p_i - M(1 - x_{ij}) \\ t_i \geq t_j + p_j - M \cdot x_{ij} \\ x_{ij} \in \{0,1\} \end{bmatrix}$$

, $\forall \{i,j\}$ che condividono le stesse risorse e non hanno precedenze predefinite

* M è detto "BIG M" e si ricava dai dati del problema!

• RELAZIONI LOGICHE:

• **AND**: $x_1, x_2, x_3 \in \{0,1\}$ e $x_3 = x_1$ AND x_2

Vincoli: $\begin{cases} x_i \in \{0,1\} & , \forall i = 1,2,3 \\ x_3 \leq x_1 \\ x_3 \leq x_2 \\ x_3 \geq x_1 + x_2 - 1 \end{cases}$

• **OR**: $x_3 = x_1$ OR x_2 $\longrightarrow$ $\begin{cases} x_i \in \{0,1\} & , \forall i = 1,2,3 \\ x_3 \geq x_1 \\ x_3 \geq x_2 \\ x_3 \leq x_1 + x_2 \end{cases}$

• **XOR**: $x_3 = x_1$ XOR x_2 $\longrightarrow$ $\begin{cases} x_i \in \{0,1\} & , \forall i = 1,2,3 \\ x_3 \leq x_1 + x_2 \\ x_1 + x_2 + x_3 \leq 2 \\ x_3 \geq x_1 - x_2 \\ x_3 \geq x_2 - x_1 \end{cases}$

- 4 PROBLEMI:

MATCHING DI CARDINALITÀ MASSIMA	STABLE SET DI CARDINALITÀ MASSIMA	EDGE COVER DI CARDINALITÀ MINIMA	VERTEX COVER DI CARDINALITÀ MINIMA
$\max \mathbb{1}^t x$ $Hx \leq \mathbb{1}$ $x \in \{0,1\}^{\lvert E \rvert}$	$\max \mathbb{1}^t y$ $H^t y \leq \mathbb{1}$ $y \in \{0,1\}^{\lvert V \rvert}$	$\min \mathbb{1}^t x$ $Hx \geq \mathbb{1}$ $x \in \{0,1\}^{\lvert E \rvert}$	$\min \mathbb{1}^t y$ $H^t y \geq \mathbb{1}$ $y \in \{0,1\}^{\lvert V \rvert}$

- **DUALITÀ:**

Matching Cardinalità max $\longleftrightarrow$ Vertex Cover cardinalità min

Stable Set Cardinalità max $\longleftrightarrow$ Edge Cover Cardinalità min

$\Rightarrow$ MATCHING $\leq$ VERTEX COVER

STABLE SET $\leq$ EDGE COVER

$\rightarrow$ Posso usarlo per trovare degli upper / lower bound!
(Basta soluzione ammissibile)

- Soluzione ammissibile del matching massimo è un LOWER BOUND per la soluzione OTTIMA del Vertex Cover di cardinalità minima, ovviamente sullo stesso grafo!

- **DUALITÀ:**

$$\begin{array}{c} \max\ c^t x \\ Ax \leq b \\ x \geq 0 \end{array} \longleftrightarrow \begin{array}{c} \min\ b^t y \\ A^t y \geq c \\ y \geq 0 \end{array}$$

con $A \in \mathbb{Z}^{m \times n}$, $c \in \mathbb{Z}^n$, $b \in \mathbb{Z}^m$, $x \in \mathbb{R}^n_+$, $y \in \mathbb{R}^m_+$.

- **TEOREMA DUALITÀ FORTE:**

$$\left[\underbrace{\begin{array}{c} c^t x \\ Ax \leq b \\ x \geq 0 \end{array}}_{\text{Ammissibile}} \leq \underbrace{\begin{array}{c} \max\ c^t x \\ Ax \leq b \\ x \geq 0 \end{array}}_{\text{Ottimo } P} = \underbrace{\begin{array}{c} \min\ b^t y \\ A^t y \geq c \\ y \geq 0 \end{array}}_{\text{Ottimo } D} \leq \underbrace{\begin{array}{c} b^t y \\ A^t y \geq c \\ y \geq 0 \end{array}}_{\text{Ammissibile}}\right]$$

→ Una soluzione ammissibile del duale è sempre un UPPER BOUND per la soluzione OTTIMA del primale!

P \ D(P)	$\exists y^*$	D è illimitato	$D = \emptyset$
$\exists x^*$	✓	✗	✗
P è illimitato	✗	✗	✓
$P = \emptyset$	✗	✓	✓

- **COMPLEMENTARY SLACKNESS:**
  - $x_j^* \cdot \left[(A_{\cdot j})^t y^* - c_j\right] = 0 \quad , \forall j = 1, \dots, n$
  - $y_i^* \cdot \left[(A_{i\cdot}) x^* - b_i\right] = 0 \quad , \forall i = 1, \dots, m$

* Se nel primale il vincolo i-esimo NON è SODDISFATTO all'uguaglianza da una soluzione x, se tale soluzione è OTTIMA $\Rightarrow y_i^* = 0$