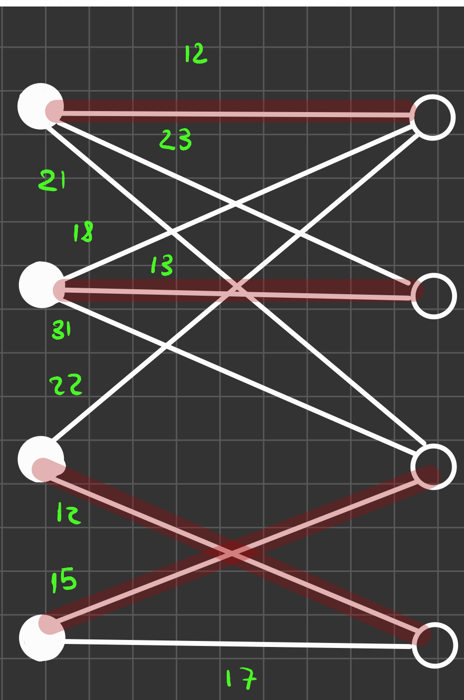
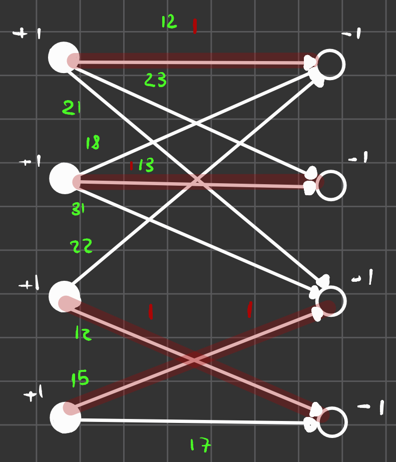
Leonardo Serilli (Mat. 274626) 4/06/21

**Exercise 1**
Thickest edges represent a perfect matching on the following graph. Prove or disprove that this matching is the minimum weight perfect matching.




// ● ∈ A
// ○ ∈ B
// M matching perfetto

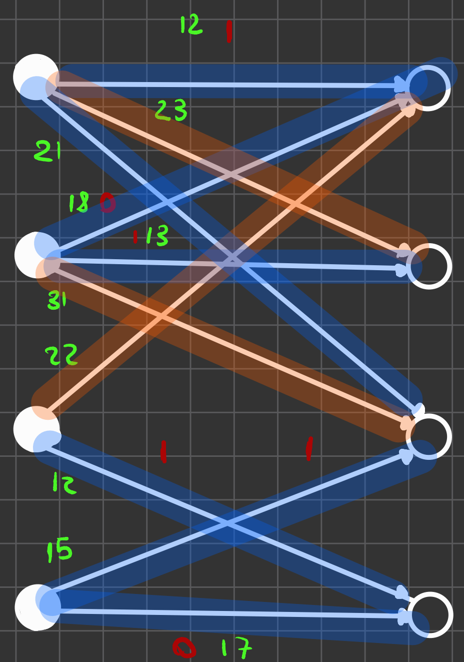
(1) Data un MATCHING PERFETTO M, costruiamoci una FT Solution {T, L} inizializzando sui nodi dell'insieme A una LABEL (supply) a +1 e sui nodi di B una LABEL (demand) a -1; orientiamo quindi gli archi da A a B inizializziamo un FEASIBLE FLOW x ponendo $x_{ij} = 1$ $\forall (i,j) \in M$.



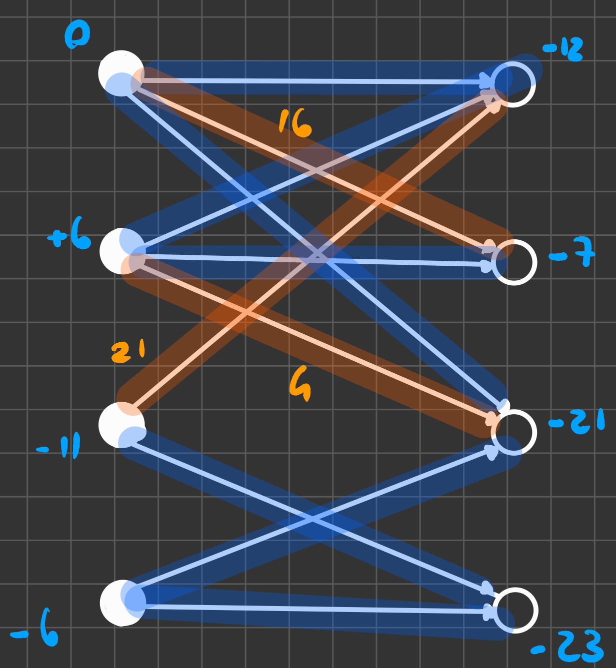
(2) Aggiungiamo a T tutti gli archi con $x_{ij} > 0$ e abbastanza con $x_{ij} = 0$ tali da rendere T un albero (iniziando da quelli di costo minimo)

(3) Aggiungiamo in L gli archi rimanenti

(4) Abbiamo ridotto il problema al MIN COST FLOW, applichiamo perciò il network simplex Algorithm



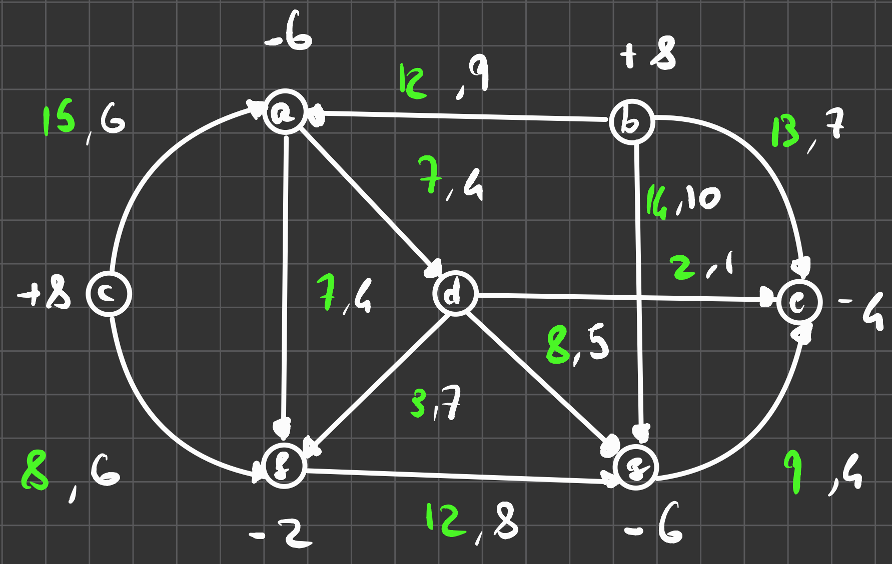
(5) Verifichiamo se l'attuale soluzione è ottima calcolando prima i POTENZIALI $y_h$, poi i COSTI RIDOTTI $\bar{c}_{ij}$




(6) La soluzione trovata rispetta la condizione di ottimalità $\bar{c}_{ij} \geq 0$

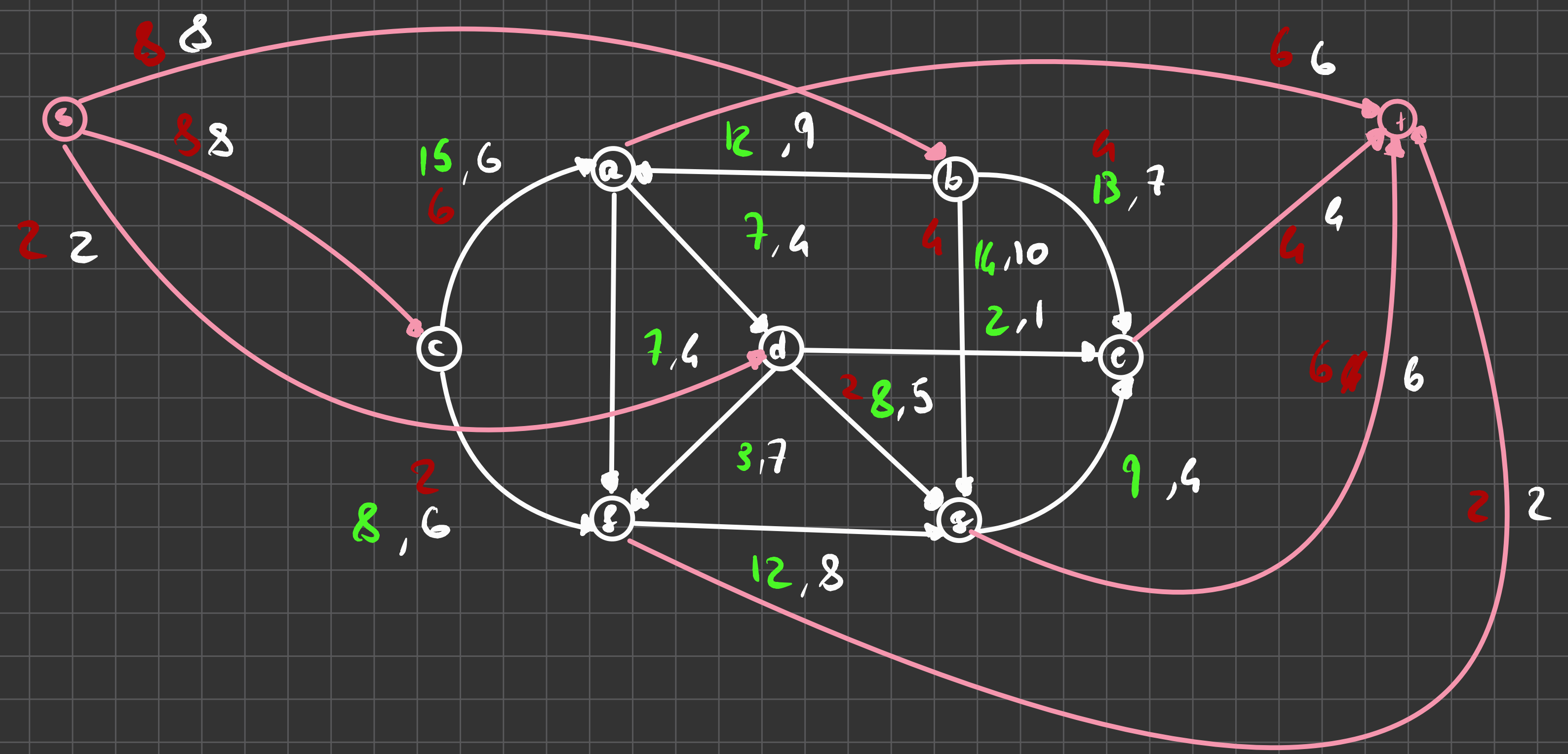
**Exercise 2**

Evaluate the min cost flow on the following graph (values on the arcs are $c_{ij}$, $u_{ij}$)



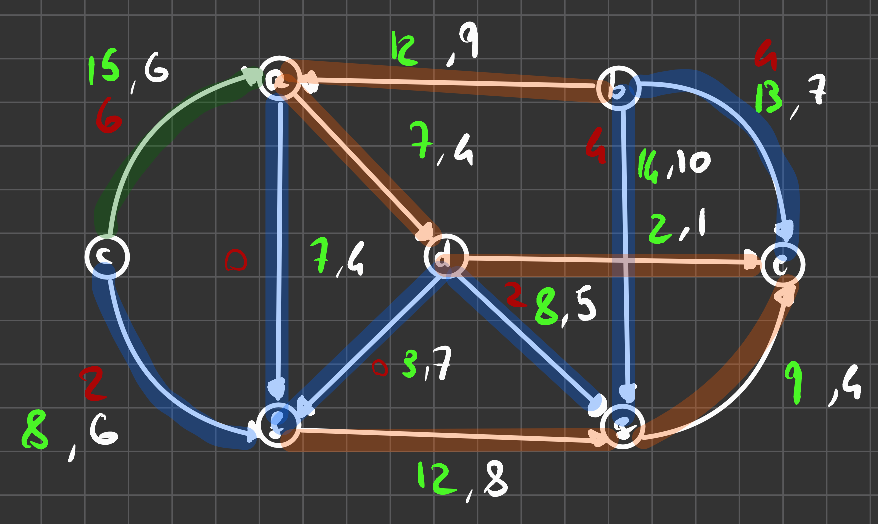

• Risolviamo il problema tramite il capacitated NETWORK FLOW ALGORITHM.

(1) Troviamo una FEASIBLE TREE SOLUTION $\{T, L, U\}$; per farlo abbiamo bisogno di inizializzare un FEASIBLE FLOW $x$ tramite l'aggiunta di due nodi $s, t$ connessi: $s$ ai nodi con LABEL $> 0$ (supply) e $t$ a quelli con LABEL $< 0$ (DOMANDA). connettiamo i due nodi $s$ e $t$ tramite archi $(s,i)$ e $(i,t)$ le cui capacity sono pari al valore assoluto della LABEL di $i$.

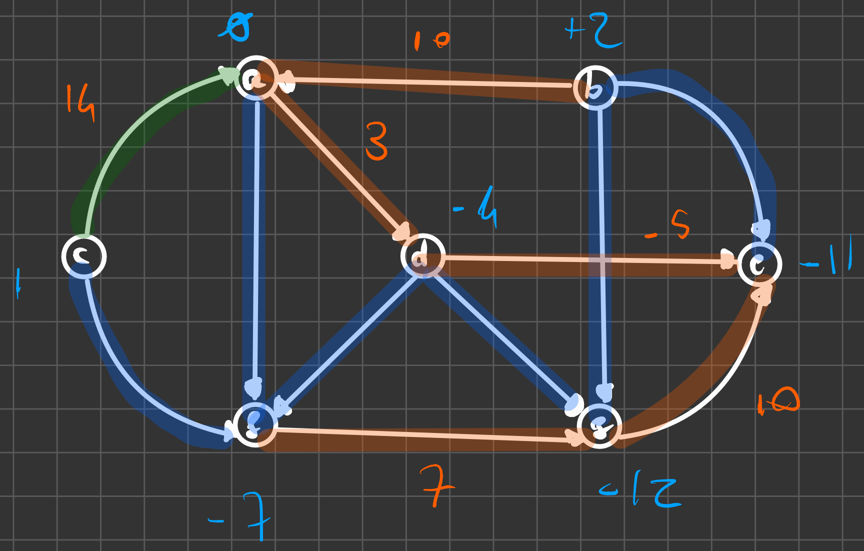


(2) Se esiste un feasible flow $X$ che satura ogni arco $(s,i)$ e $(i,t)$ possiamo ricavarci $\{T, L, U\}$:

- mettiamo in $U$ ogni arco $(i,j)$ t/c $x_{ij} = u_{ij}$
- mettiamo in $T$ ogni arco $(i,j)$ t/c $0 \le x_{ij} < u_{ij}$, dando però la priorità agli archi con $x_{ij} > 0$, aggiungendo quelli con $x_{ij} = 0$ solo per rendere $T$ un albero
- mettiamo in $L$ ogni arco $(i,j)$ t/c $x_{ij} = 0$, a meno che $(i,j)$ non sia già stato aggiunto a $T$



(3) Calcoliamo i potenziali $y_h = -\sum_{(i,j)\in FORWARD} c_{ij} + \sum_{(i,j)\in REVERSE} c_{ij}$ $\forall (i,j) \in PATH_T(Root, h)$, e i costi ridotti $\overline{c_{ij}} = c_{ij} - y_i + y_j$ della soluzione $\{T, L, U\}$, al fine di verificarne l'ottimalità



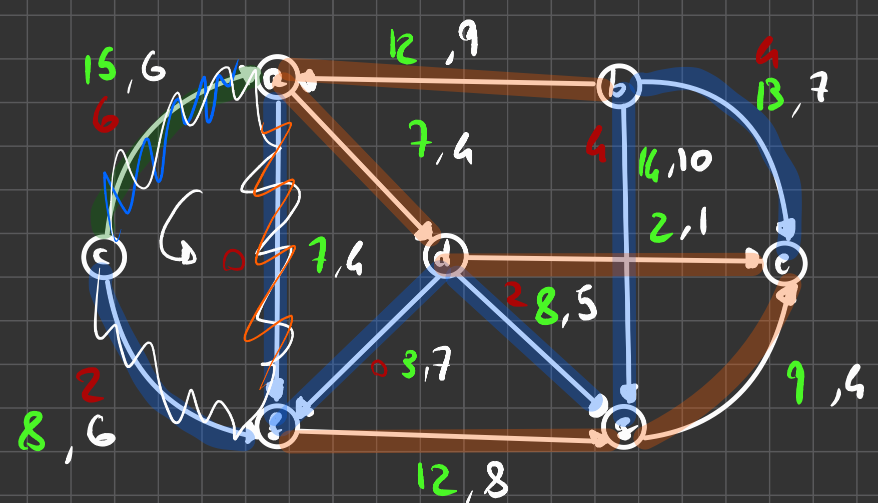

(4) La soluzione 2 non è ottima perché non soddisfa la condizione $\begin{cases} \overline{c_{ij}} \le 0 & \forall (i,j) \in U \\ \overline{c_{ij}} \ge 0 & \forall (i,j) \in L \end{cases}$

(5) PIVOTING: Preso un arco (i,j) che viola la condizione di ottimalità, aggiungendolo alla soluzione, troviamo il ciclo C generato, orientiamo il ciclo nel verso opposto a (i,j) se $(i,j)\in U$, nello stesso se invece $(i,j) \in L$ e calcoliamo il valore di massimo cambiamento nel ciclo $\varepsilon$

(6) Incrementiamo il flow di $\varepsilon$ sugli archi forward rispetto il ciclo e lo decrementiamo sugli archi con direzione opposta

• Verifichiamo qual è il miglior arco da aggiungere a T supponendo di aggiungerlo e calcolando $\varepsilon \cdot |\overline{c_{ij}}|$; L'arco con il massimo valore $\varepsilon \cdot |\overline{c_{ij}}|$ è la miglior scelta (perché $z' = z - \varepsilon|\overline{c_{ij}}|$)

① . SUPPONIAMO DI AGGIUNGERE (c, a)



// CICLO

// ARCO AGGIUNTO

// ARCO DA RIMUOVERE

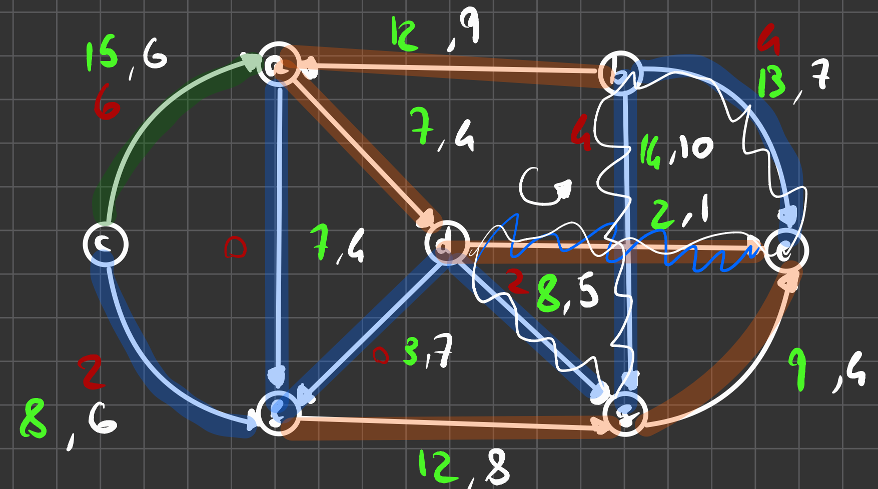
$\Rightarrow \varepsilon = 0 \Rightarrow z' = z$

// CICLO

// ARCO AGGIUNTO

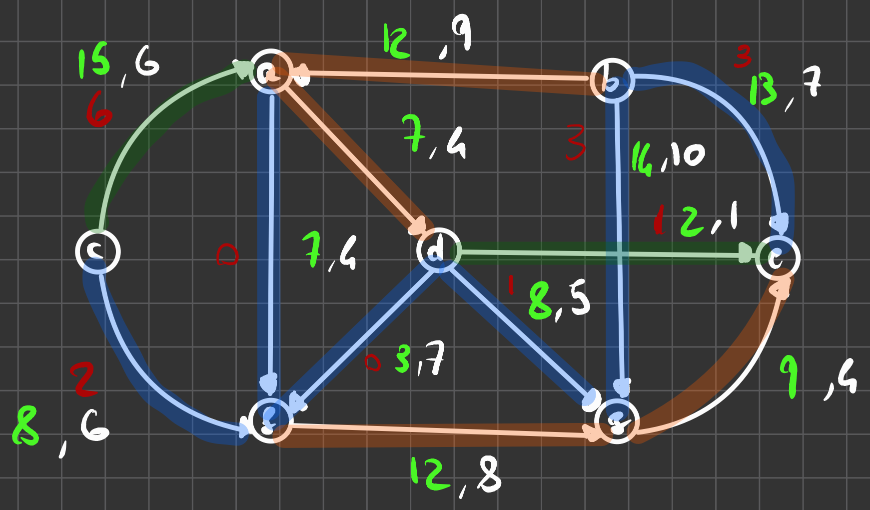
// ARCO DA RIMUOVERE

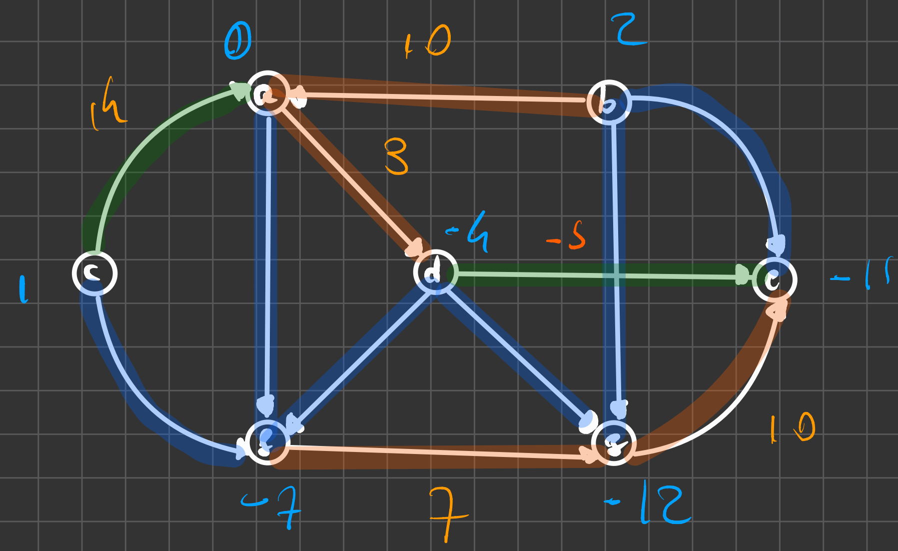
② . SUPPONIAMO DI AGGIUNGERE (d, e)



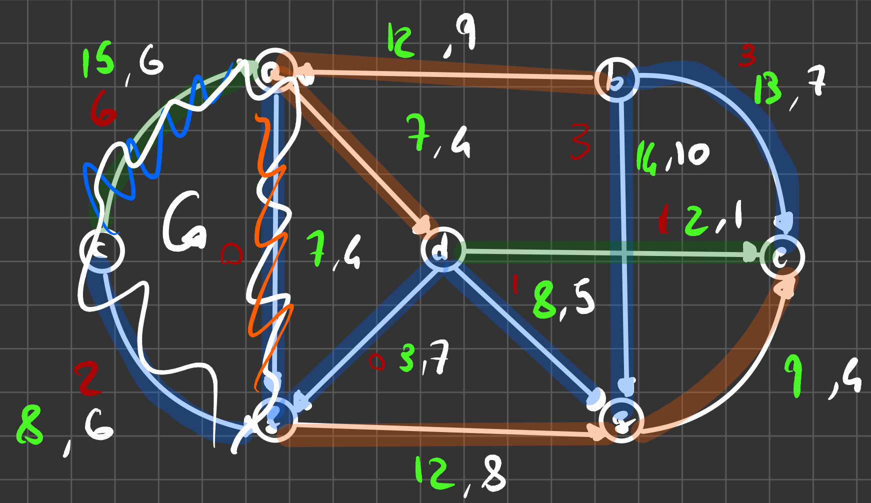
$\Rightarrow \varepsilon = 1 \Rightarrow z' = z - 5$ ✓

⇒ La soluzione migliore è la ②, proseguiamo da lì, notiamo che l'arco (d, e) viene saturato (quindi aggiunto ad U)

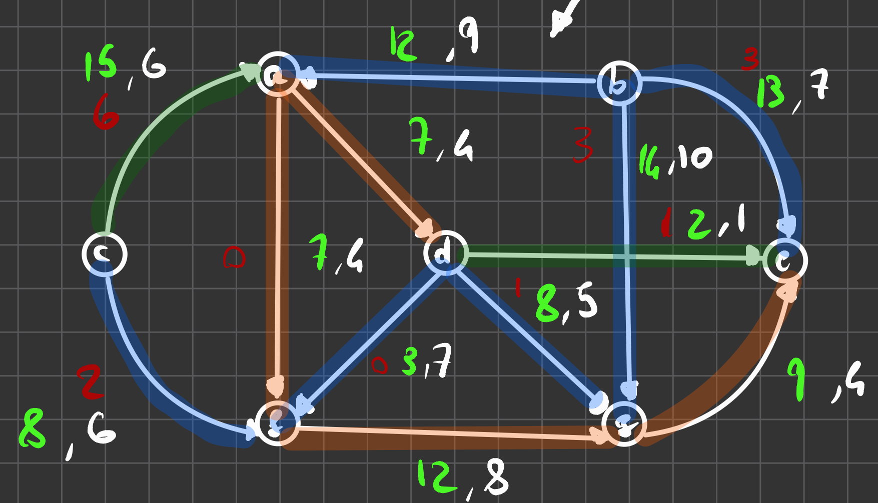


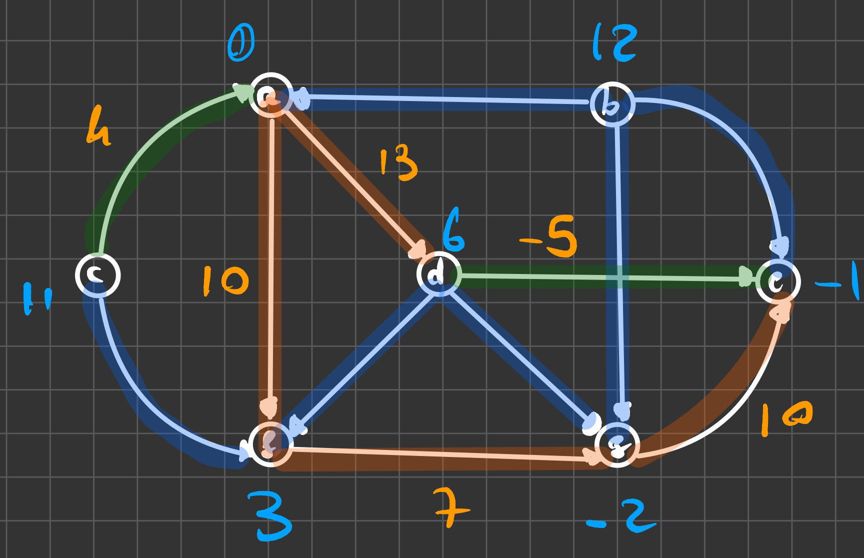




//DEGENERATE ITERATION

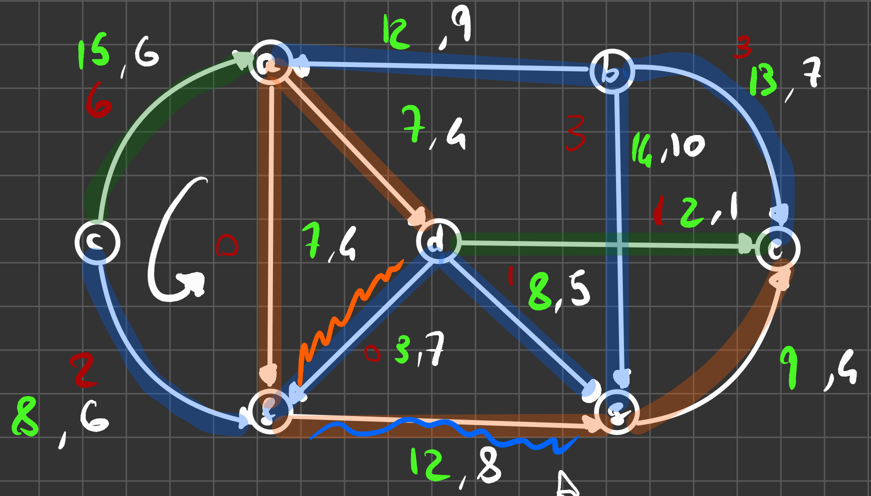



ADD (a,b) TO T

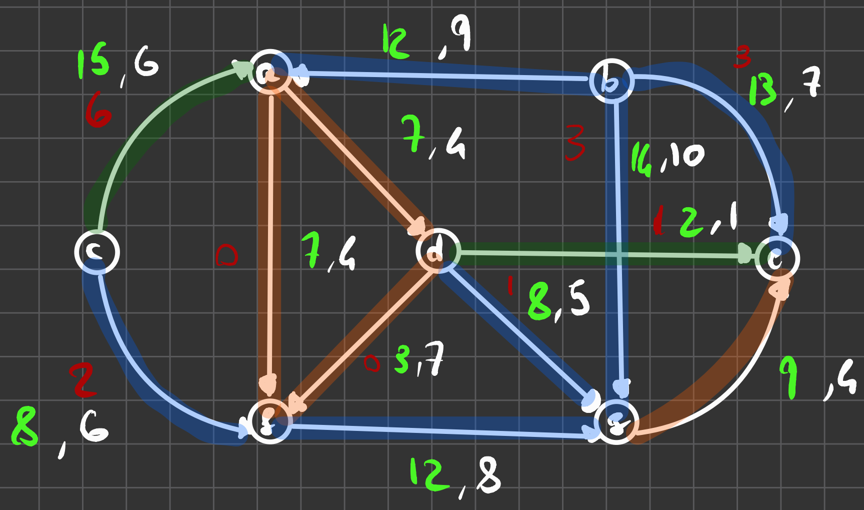


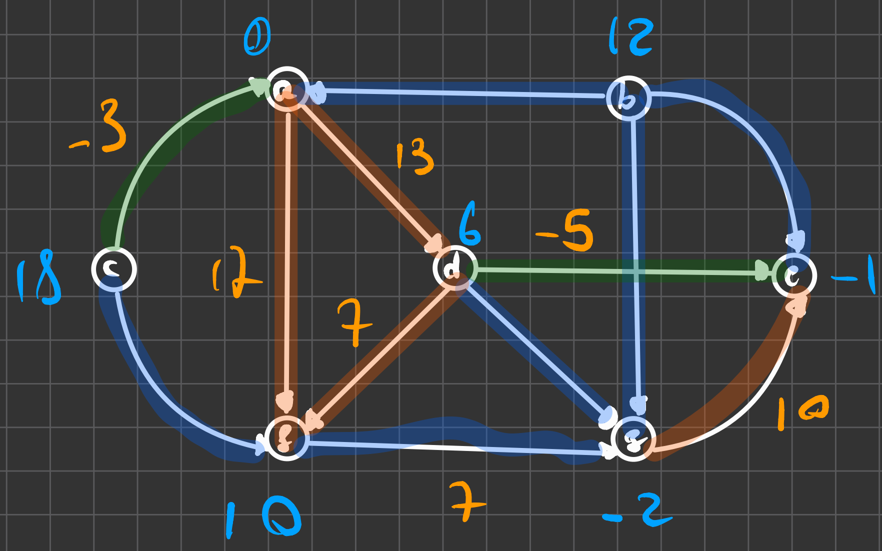




//DEGENERATE ITERATION




ADD (f,g) to T




=> LA SOLUZIONE È ORA OTTIMA e VALE:

$$z = 3\cdot 13 + 3\cdot 14 + 1\cdot 8 + 2\cdot 8 + 6\cdot 15 + 2\cdot 1 =$$

$$= 39 + 42 + 8 + 16 + 90 + 2 =$$

$$= \boxed{197}$$

**Exercise 3**

A company has 2 types of employees: part-time (5 hours work shift) and full-time (7 hours work shift). An employee can start his/her work shift at {8:00, 9:00, 12:00, 13:00}. For each hourly slot of a day the request of personnel is the following:

| 8:00 9:00 | 9:00 10:00 | 10:00 11:00 | 11:00 12:00 | 12:00 13:00 | 13:00 14:00 | 14:00 15:00 | 15:00 16:00 | 16:00 17:00 | 17:00 18:00 | 19:00 20:00 |
|---|---|---|---|---|---|---|---|---|---|---|
| 12 | 14 | 18 | 14 | 10 | 12 | 8 | 6 | 5 | 5 | 3 |

Given a cost of 230 Euro/day for part time employees and 340 Euro/day for full time employees, describe the model to find the mix of employees that fulfills the request minimizing the total daily cost.

**Bonus question**
Explain how to use the same model to check if it convenient to allow full time employees start working at 11:00.

- Lo slot di orario '18-19' non è richiesto dal problema
- Alcuni constraint sono ridondanti e possono essere rimossi

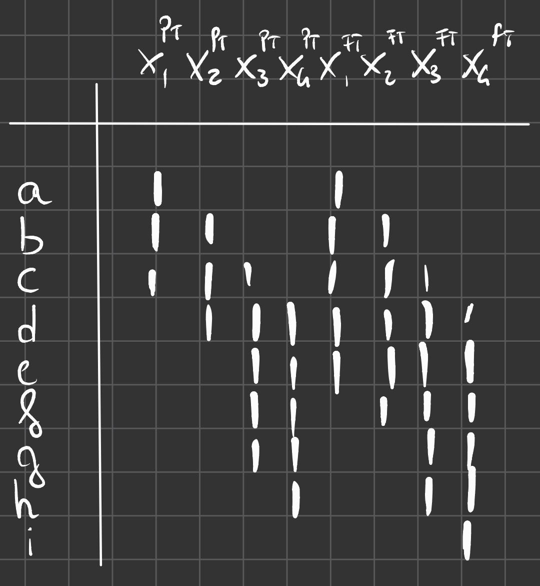
| | |
|---|---|
| $\boxed{8\text{-}9}$ | $x_1^{PT} + x_1^{FT} \geq 12$ |
| ~~$\boxed{9\text{-}10}$~~ | ~~$x_1^{PT} + x_2^{PT} + x_1^{FT} + x_2^{FT} \geq 14$~~ |
| 10-11 | $x_1^{PT} + x_2^{PT} + x_1^{FT} + x_2^{FT} \geq 18$ |
| ~~11-12~~ | ~~$x_1^{PT} + x_2^{PT} + x_1^{FT} + x_2^{FT} \geq 14$~~ |
| $\boxed{12\text{-}13}$ | $x_1^{PT} + x_2^{PT} + x_3^{PT} + x_1^{FT} + x_2^{FT} + x_3^{FT} \geq 10$ |
| $\boxed{13\text{-}14}$ | $x_2^{PT} + x_3^{PT} + x_4^{PT} + x_1^{FT} + x_2^{FT} + x_3^{FT} + x_4^{FT} \geq 12$ |
| 14-15 | $x_3^{PT} + x_4^{PT} + x_1^{FT} + x_2^{FT} + x_3^{FT} + x_4^{FT} \geq 8$ |
| 15-16 | $x_3^{PT} + x_4^{PT} + x_2^{FT} + x_3^{FT} + x_4^{FT} \geq 6$ |
| 16-17 | $x_3^{PT} + x_4^{PT} + x_3^{FT} + x_4^{FT} \geq 5$ |
| 17-18 | $x_4^{PT} + x_3^{FT} + x_4^{FT} \geq 5$ |
| 19-20 | $x_4^{FT} \geq 3$ |

$\Rightarrow$ FORMA $Ax \geq b$

- La funzione obiettivo è

$$\boxed{\min \sum_{i=1}^{4} 230\, x_i^{PT} + \sum_{i=1}^{4} 340\, x_i^{FT}}$$

- Verifichiamo se la matrice **A** ha 'consecutive ones prop.'

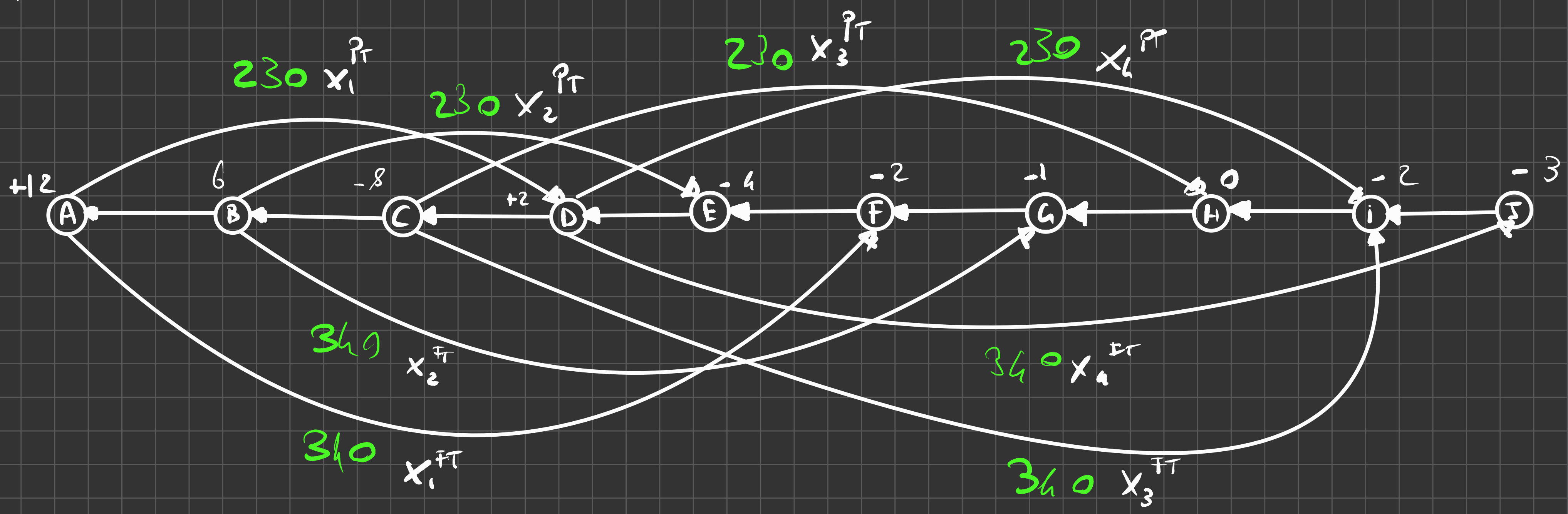



$\triangleright$ SODDISFA 'CONS. 1s PROP'

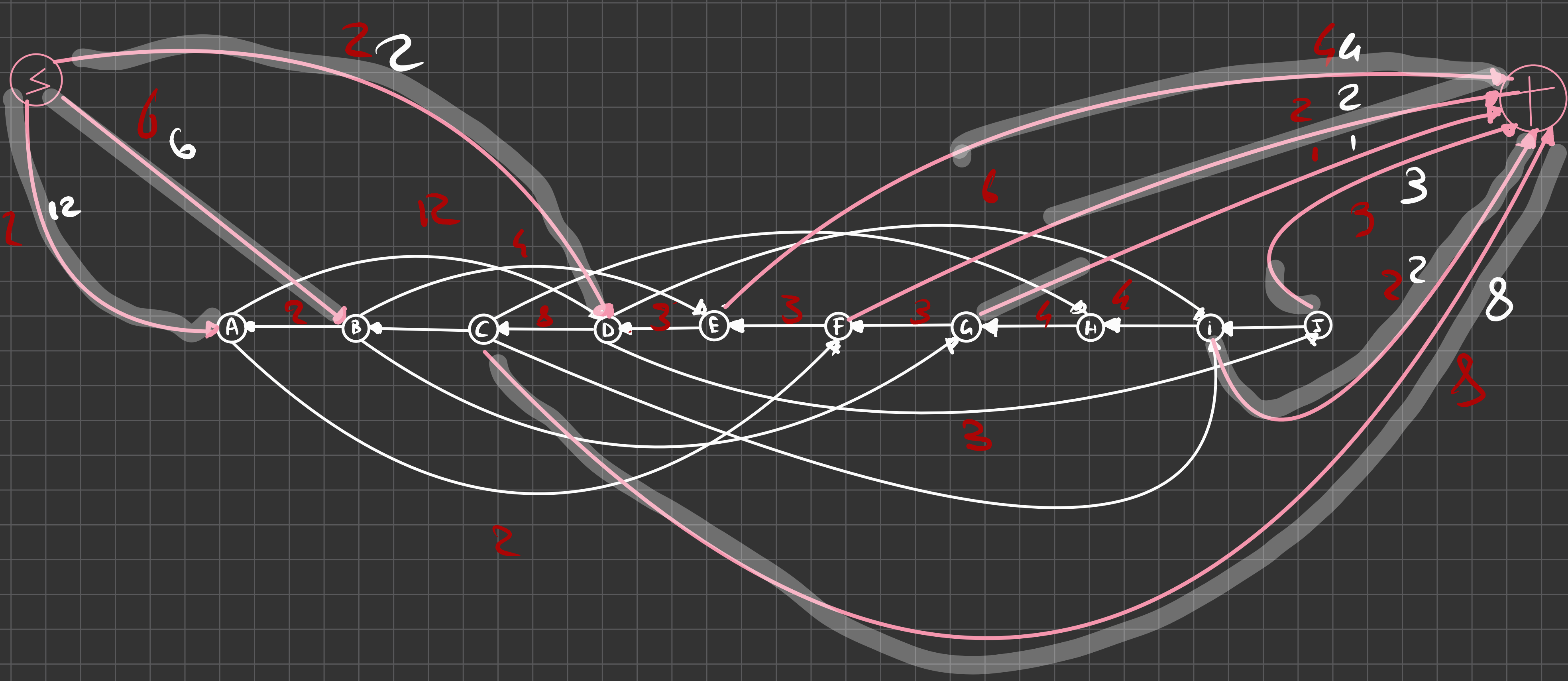
• Trasformiamo la matrice in una NETWORK MATRIX

| | $x_1^{PT}$ | $x_2^{PT}$ | $x_3^{PT}$ | $x_4^{PT}$ | $x_1^{FT}$ | $x_2^{FT}$ | $x_3^{FT}$ | $x_4^{FT}$ | $y_a$ | $y_b$ | $y_c$ | $y_d$ | $y_e$ | $y_f$ | $y_g$ | $y_h$ | $y_i$ | b |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | 1 | | | | 1 | | | | -1 | | | | | | | | | 12 |
| b | 1 | 1 | | | 1 | 1 | | | | -1 | | | | | | | | 18 |
| c | 1 | 1 | 1 | | 1 | 1 | 1 | | | | -1 | | | | | | | 10 |
| d | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | -1 | | | | | | 12 |
| e | | | 1 | 1 | 1 | 1 | 1 | 1 | | | | | -1 | | | | | 8 |
| f | | | 1 | 1 | | 1 | 1 | 1 | | | | | | -1 | | | | 6 |
| g | | | 1 | 1 | | | 1 | 1 | | | | | | | -1 | | | 5 |
| h | | | | 1 | | | 1 | 1 | | | | | | | | -1 | | 5 |
| i | | | | | | | | 1 | | | | | | | | | -1 | 3 |
| J | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

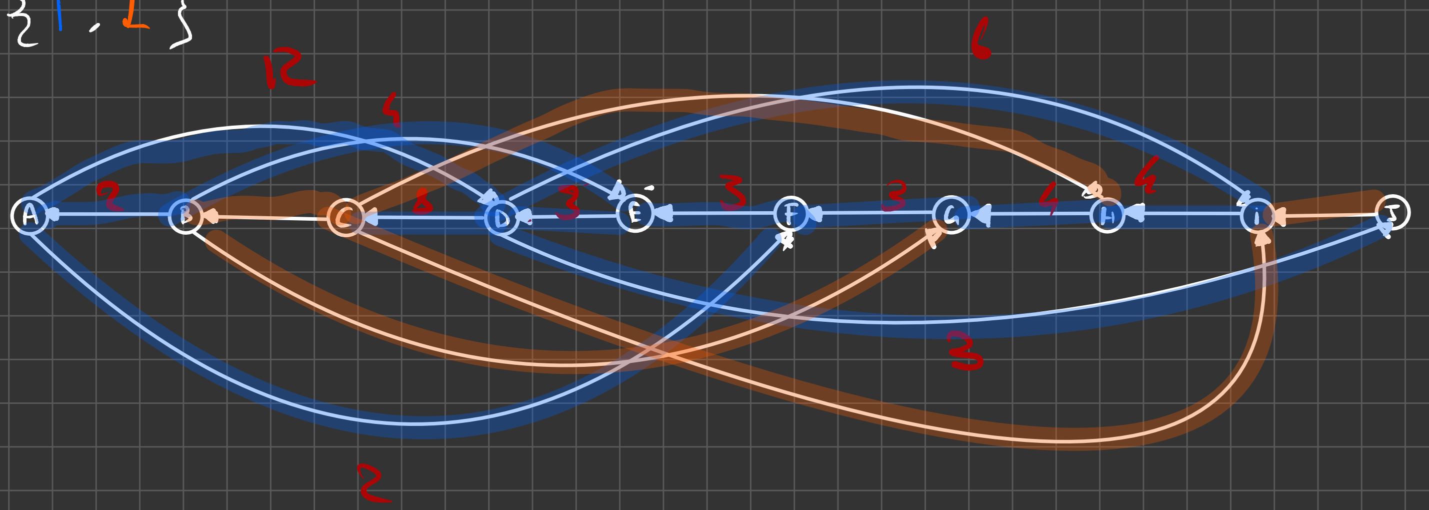
| | $x_1^{PT}$ | $x_2^{PT}$ | $x_3^{PT}$ | $x_4^{PT}$ | $x_1^{FT}$ | $x_2^{FT}$ | $x_3^{FT}$ | $x_4^{FT}$ | $y_a$ | $y_b$ | $y_c$ | $y_d$ | $y_e$ | $y_f$ | $y_g$ | $y_h$ | $y_i$ | b |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | | | | 1 | | | | -1 | | | | | | | | | 12 |
| B | | 1 | | | | 1 | | | | -1 | | | | | | | | 6 |
| C | | | 1 | | | | 1 | | | | -1 | | | | | | | -8 |
| D | -1 | | | 1 | | | | 1 | | | | -1 | | | | | | 2 |
| E | | -1 | | | | | | | | | | | -1 | | | | | -4 |
| F | | | | | -1 | | | | | | | | | -1 | | | | -2 |
| G | | | | | | -1 | | | | | | | | | -1 | | | -1 |
| H | | | -1 | | | | | | | | | | | | | -1 | | 0 |
| I | | | | -1 | | | -1 | | | | | | | | | | -1 | -2 |
| J | | | | | | | | -1 | | | | | | | | | 1 | -3 |




• Troviamo un flow ammissibile per ricavare una FT Solution {T, L}




• FTS := {T, L}




• ORA IL PROBLEMA È STATO RIDOTTO A UN MIN COST FLOW PROBLEM