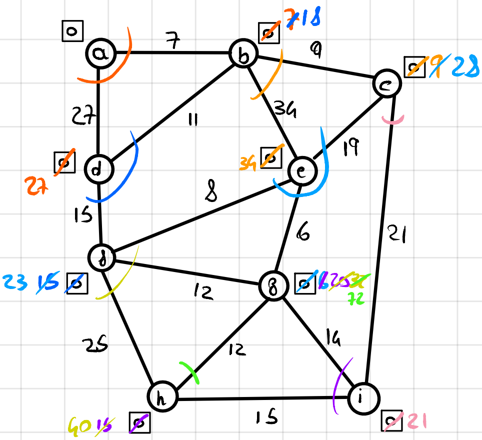
Seralli Leonardo (274426) 14/05

Exercise 1

Evaluate the minimum cut on the following graph

(1) Troviamo Legal ordering ovvero dato $G=(V,A)$

- > Inizializzo $O = \{\emptyset\}$
- > Assegno una label $l_i = 0 \ \forall i \in N$
- > Finché $|O| \neq |V|$:
  - > Prendo il nodo i con massima label l_i
  - > Aggiorno $l_j = l_j + u((i,j))$ per ogni vicino j di i tale che $j \notin O$
  - > Aggiungo i al legal ordering O



$O = \{a, d, b, e, c, i, f, h, g\}$

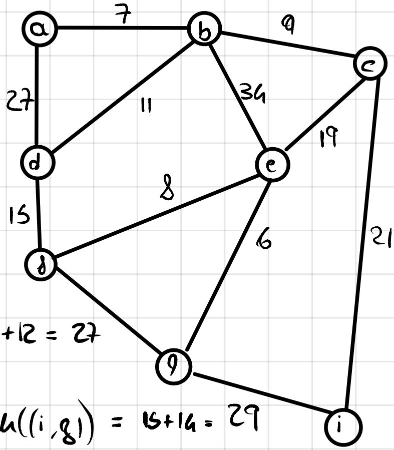
(2) Troviamo il valore del minimo taglio che non separi i nodi i e j, dove $\{i,j\}$ sono gli ultimi due nodi aggiunti al legal ordering nello step 1, perché:

- Dato un LEGAL ORDERING $O = (a_1, a_2, \ldots, i, j) \Rightarrow \lambda(G, j, i) = u(\delta(j)) = l_j$

Nel nostro caso $\{i,j\} = \{h,g\}$, segue quindi $\Rightarrow$ $\boxed{\lambda(G, g, h) = u(\delta(g)) = 72}$

(3) Costruiamo il grafo G_{ij} identificando i nodi i,j descritti allo step 2

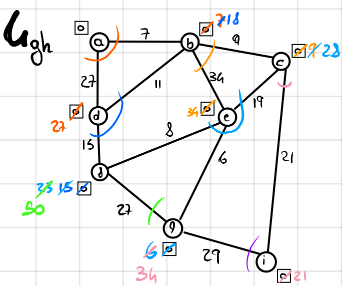
$G_{gh} = (V_{gh}, N_{gh}) \rightarrow$



$w((g,h)) + w((f,g)) = 25+12 = 27$

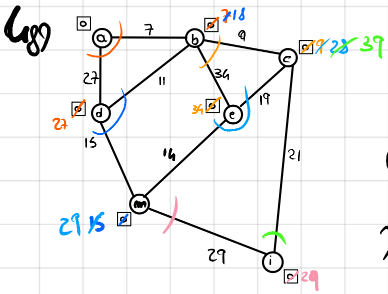
$w((i,h)) + w((i,g)) = 15+14 = 29$

(4) Iteriamo gli step sul grafo G_{is} ottenuto allo step 3 finché $|V_{is}|>1$. Al termine dell'Algoritmo il valore del global min cut $\lambda(G)$ sarà pari al minimo $\lambda(G,i,j)$ per ogni coppia $\{i,j\}$ identificata nelle esecuzioni dello step 3



$O = \{a, d, b, e, c, i, g, f\}$

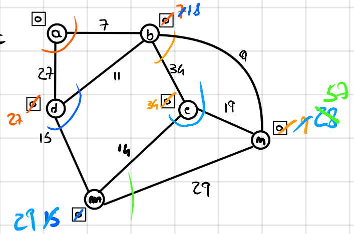
$\lambda(G, f, g) = w(\delta(f)) = 50$



$O = \{a, d, b, e, m, i, c\}$

$\lambda(G, i, c) = w(\delta(c)) = 39$

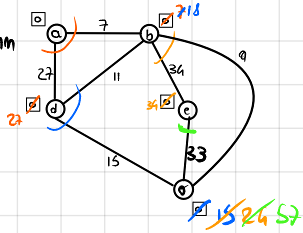
G_{ic}



$$O = \{a, d, b, e, m, n\}$$

$$\lambda(G, m, n) = \mu(\delta(n)) = 57$$

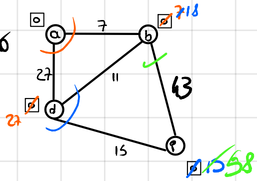
G_{mn}



$$O = \{a, d, b, e, \sigma\}$$

$$\lambda(G, e, \sigma) = \mu(\delta(\sigma)) = 57$$

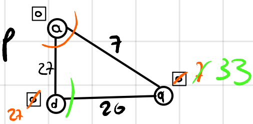
$G_{e\sigma}$



$$O = \{a, d, b, \varphi\}$$

$$\lambda(G, b, \varphi) = \mu(\delta(\varphi)) = 58$$

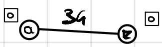
$G_{b\varphi}$



$$O = \{a, d, q\}$$

$$\lambda(G, d, q) = \mu(\delta(q)) = 33$$

G_{dq}

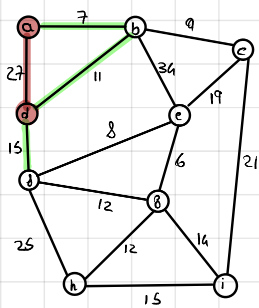


$$O = \{a, z\}$$

$$\lambda(G, a, z) = \mu(\delta(z)) = 34$$

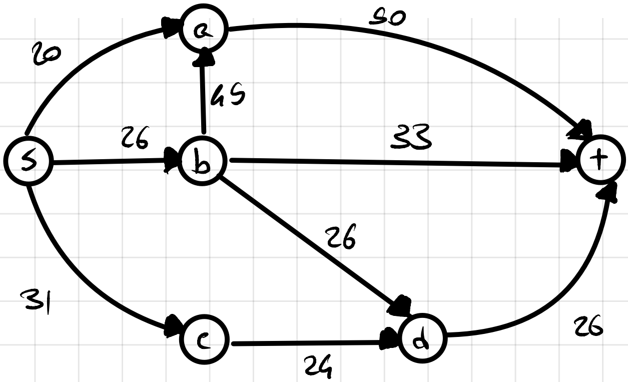
$\Rightarrow$

Il minimo $\lambda(G,i,j)$ è $\lambda(G,d,q) = 33$ perciò il min GLOBAL CUT comprende i nodi $\{a,d\}$, e il suo valore è dato dalla somma delle capacità sugli archi che separano l'insieme $\{a,d\}$ dal resto del grafo



Exercise 2

Given the following graph $G = (N, A)$ evaluate the maximum flow $f_x(s)$ from s to t by the preflow-push algorithm.

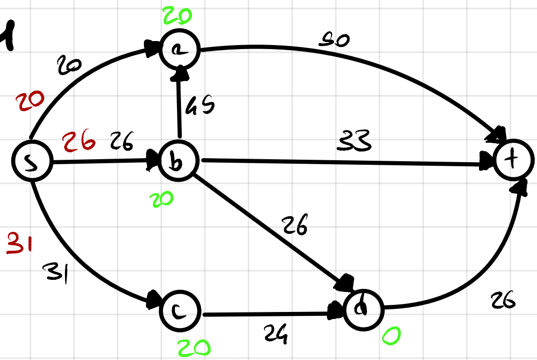


① Inizializziamo Preflow x e Eccessi

- $x_{ij} = \begin{cases} u_{ij} & \text{se } (i,j) \in \delta^-(s) \\ 0 & \text{altrimenti} \end{cases}$

- $e_i = \sum_{(i,j)\in\delta^-(i)} x_{ij} - \sum_{(i,j)\in\delta^+(i)} x_{ij}$

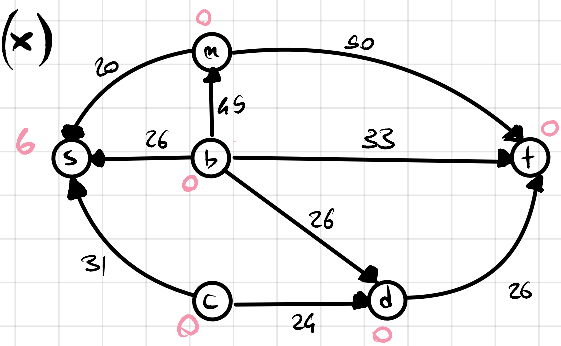
G



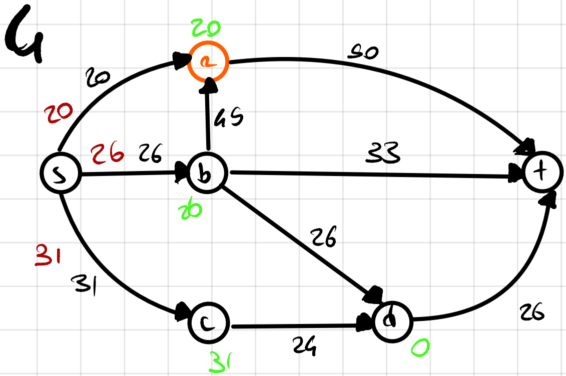
② Costruiamo G(x) e inizializziamo le labels d :

- $d_i = \begin{cases} n & \text{se } i=s \\ 0 & \text{altrimenti} \end{cases}$

G(x)



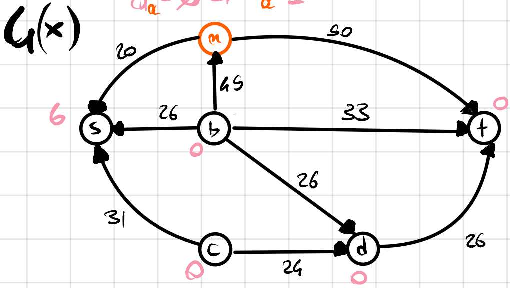
③ Selezioniamo un NODO ATTIVO i $(i \mid e_i > 0)$ scegliendo quello con massima label d, in questo caso prendiamo il nodo a



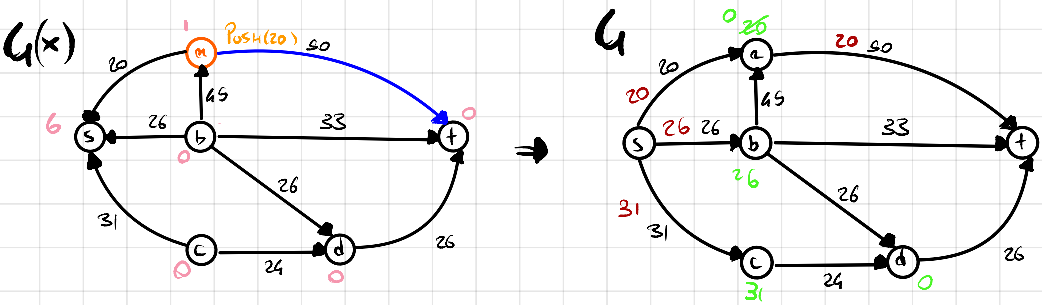
⑥ Dato che non esiste un arco ammissibile $((i,s) \mid d_i = d_s + 1)$ per il nodo selezionato allo step 3, effettuiamo il Relabel di a, ovvero:

$d_a = d_s + 1 \mid d_s =$ minima label nel vicinato di a

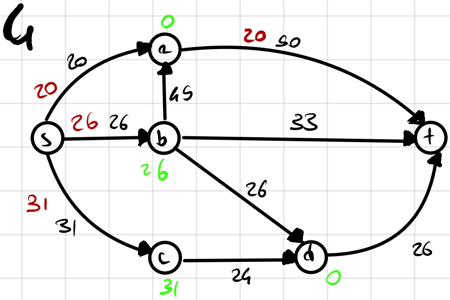
$d_a = \cancel{0} \rightarrow d_a = 1$

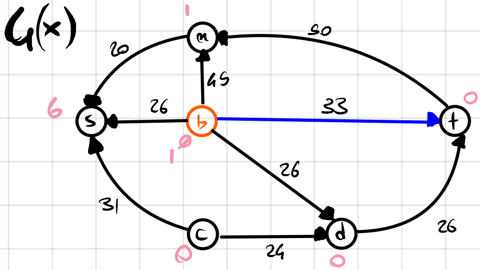


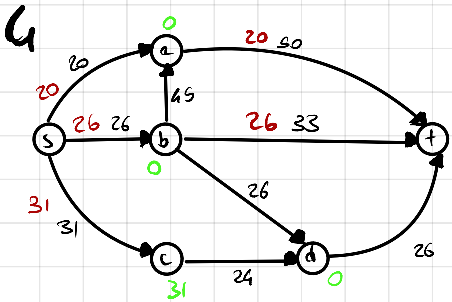
⑥ Per costruzione, dopo il Relabel esiste almeno un arco ammissibile per a, nei prossimi step l'algoritmo effettuerà una PUSH su ognuno di essi fino ad azzerare l'eccesso e_i o a saturare gli archi ammissibili

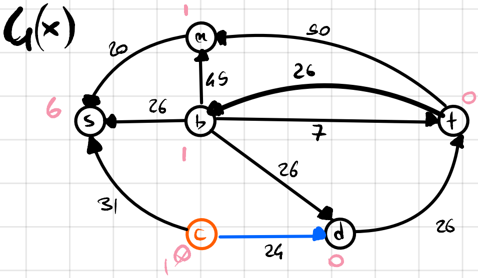


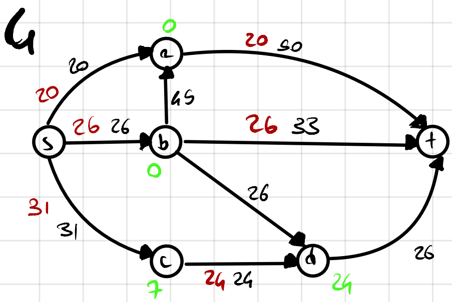
5. Ripetiamo gli step da 2 a 4 finché gli eccessi non diventano nulli su ogni nodo che non sia s o t.

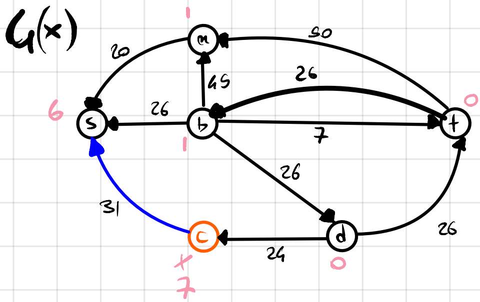


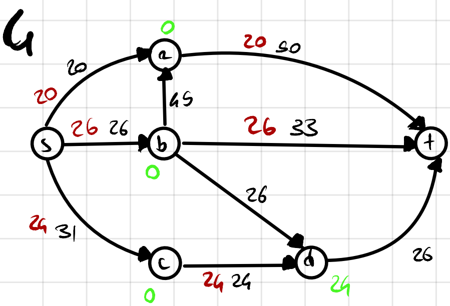


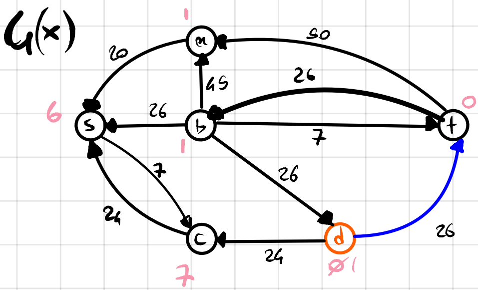




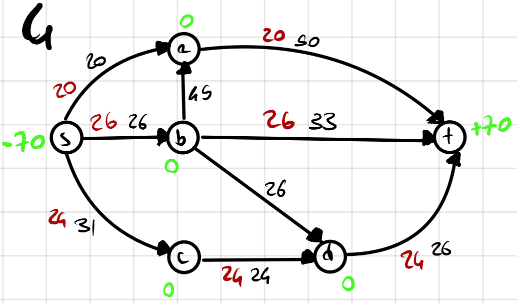








4



→ $\varphi^{*}_{x} = 70$

Exercise 3
Given the following matrix:

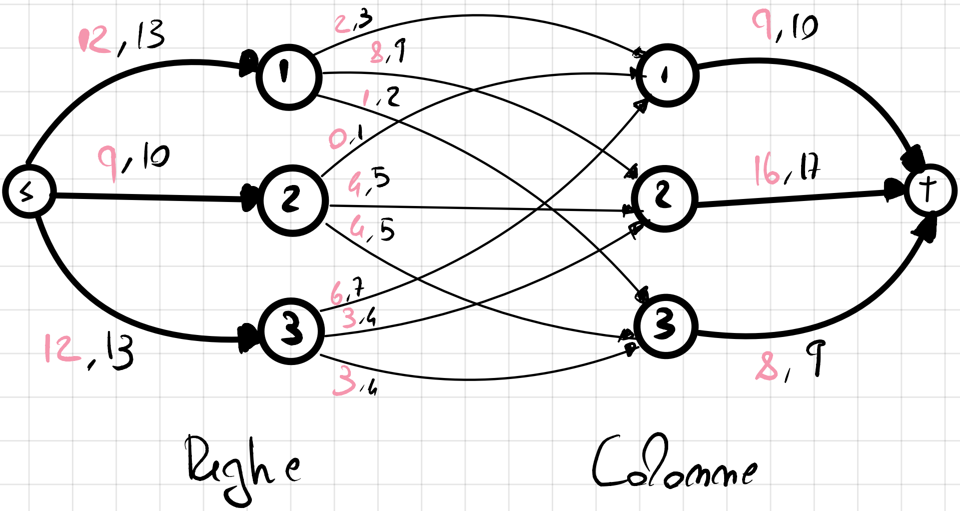
	j=1	j=2	j=3	
i=1	2.31	8.78	1.18	12.27
i=2	0.75	4.43	4.31	9.49
i=3	6.23	3.28	3.14	12.62
	9.29	16.49	8.63	

Let

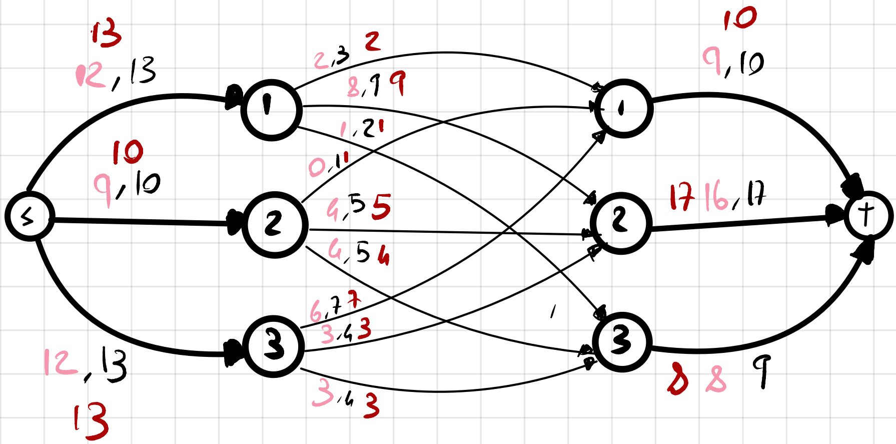
$$r_i = \sum_{j=1}^{4} a_{ij}, \text{ for } i \in 1, \ldots, 3$$

$$c_j = \sum_{i=1}^{3} a_{ij}, \text{ for } j \in 1, \ldots, 4$$

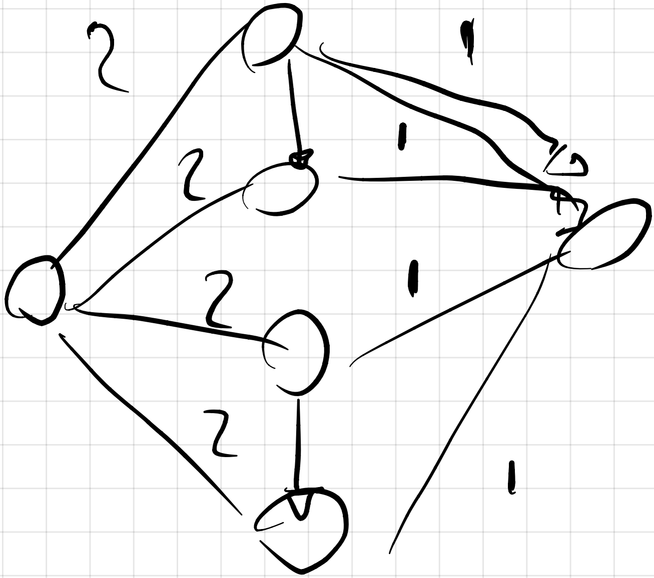
Round each element a_{ij}, r_i and c_j **up** or **down** to integer so that the sum of the rounded elements in each row (column) equals the rounded row (column) sum of the original values.



Soluzione:



2	9	1	13
1	5	4	10
7	3	3	12
10	17	8	


2
1
1
2
1
2
2
1