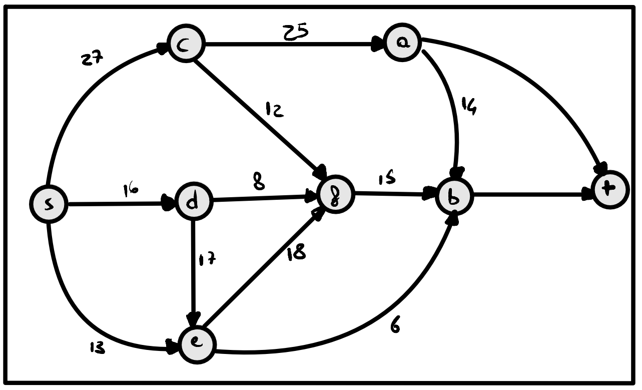
PUSH RELABEL

PUSH LABEL ALGORITHM

INIZIALIZZA: Preflow X, Labels d, Eccessi e

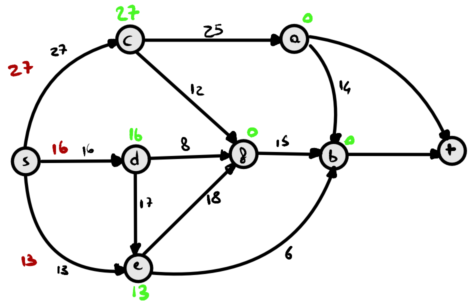
WHILE (X NOT FEAS FLOW)

- Scegli NODO ATTIVO i (con $e>0$) su $G(x)$
- WHILE ($\exists$ ARCO AMMISSIBILE (i,j) t.c. $d_i = d_j + 1$)
  - PUSH (i,j)
- IF (i è ancora attivo)
  - RELABEL(i)



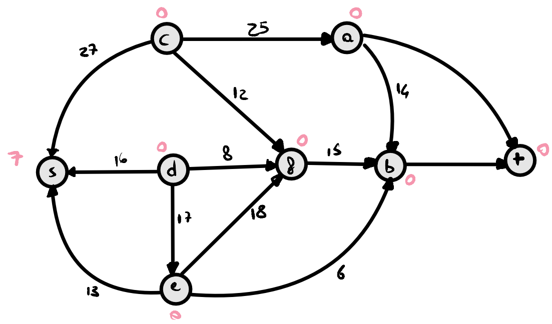
① Inizializziamo Preflow x e Eccessi

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

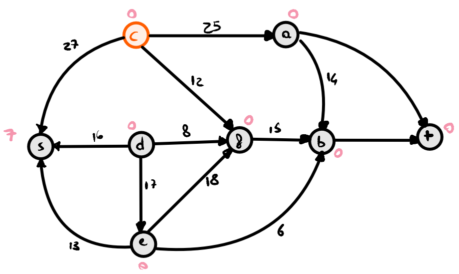


② Costruiamo $G(x)$ e inizializziamo le labels d:

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

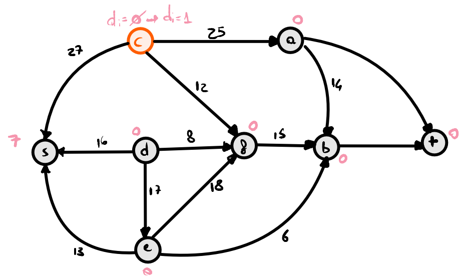


③ Selezioniamo un NODO ATTIVO i ($i \mid e_i > 0$) scegliendo quello con massima label d

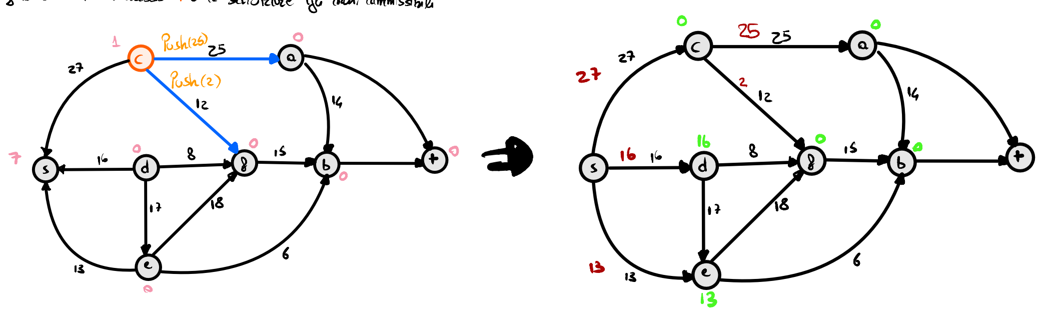


④ Dato che non esiste un arco ammissibile ($(i,j) \mid d_i = d_j + 1$) per il nodo i selezionato effettuiamo il Relabel di i, ovvero:

$d_i = d_j + 1 \mid d_j$ = minima label nel vicinato di i



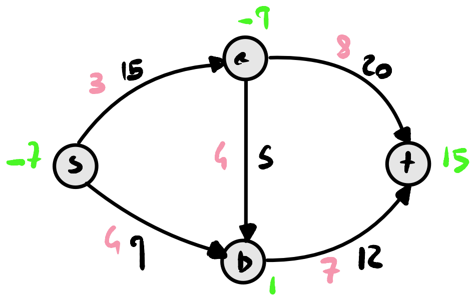
⑥ Per costruzione, dopo il Relabel esiste almeno un arco ammissibile per i, nei prossimi step l'algoritmo effettuerà una PUSH su ognuno di essi fino ad azzerare l'eccesso e_i o a saturare gli archi ammissibili



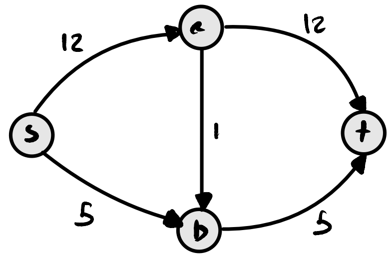
FLOW CON LOWER BOUNDS

① Inizializziamo gli eccessi in $G=(N,E)$:

$e_k = \sum_{(i,j)\in\delta^-(k)} l_{ij} - \sum_{(i,j)\in\delta^+(k)} l_{ij} \quad \forall k \in N$



② Effettuiamo lo scaling delle capacity u_{ij} costruendo un grafo G' in cui $u'_{ij} = u_{ij} - l_{ij}$



③ Introduciamo in G':

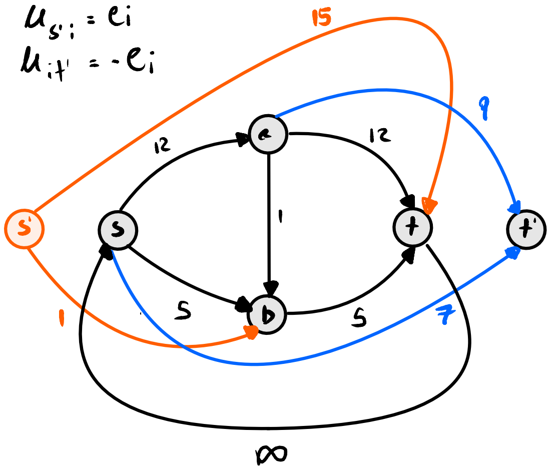
- un arco $(s,t) \mid u_{st} = \infty$
- due nodi s', t'
- un arco (s',i) $\forall i \in E$ t/c $e_i > 0$
- un arco (i,t') $\forall i \in E$ t/c $e_i < 0$

Impostiamo:

$u_{s'i} = e_i$

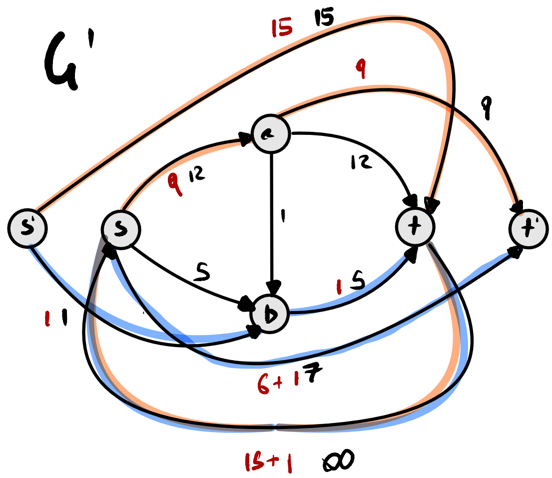
$u_{it'} = -e_i$

G'



④ Se esiste un FEASIBLE (s',t')-flow che satura ogni arco $(i,j) \mid (i,j)\in\delta^+(s')$ OR $(i,j)\in\delta^-(t')$ allora il grafo G ammette un FEASIBLE (s,t)-flow

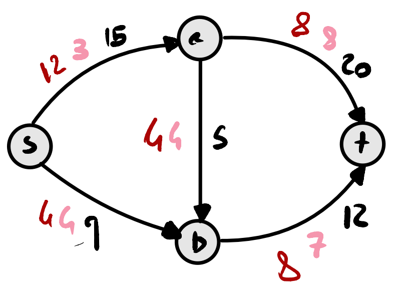
G'



⑤ Se il problema ammette feasible (s,t)-flow x inizializziamolo in G assegnando:

$$x_{ij} = l_{ij} + x'_{ij}$$

G



⑥

troviamo ora un flow ottimo

...

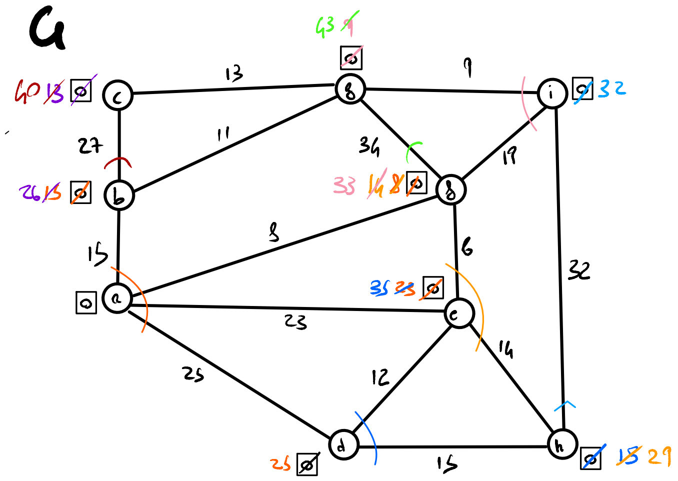
$$f_x = u(\delta(R)) - l(\delta(\bar{R}))$$

GLOBAL MIN CUT

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

> Inizializzo $O = \{\emptyset\}$
> Assegno una label $l_i = 0 \quad \forall i \in N$
> 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

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



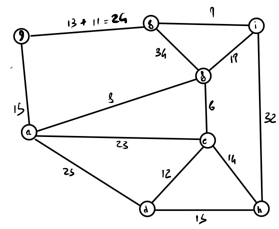
(2) Troviamo il valore del minimo taglio che non separi i nodi i e j, dove $\{i,j\}$ sono gli ultimi due nodi aggiunti al legal ordering nello step 1, perché:

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

$$\lambda(G,c,b) = u(\delta(c)) = 40$$

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

$G_{cb} = (V_{cb}, E_{cb})$



(4) Iteriamo gli step sul grafo G_{ij} ottenuto al punto (3) finché $|V_{ij}| > 1$.
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 iterazioni dello step (3)

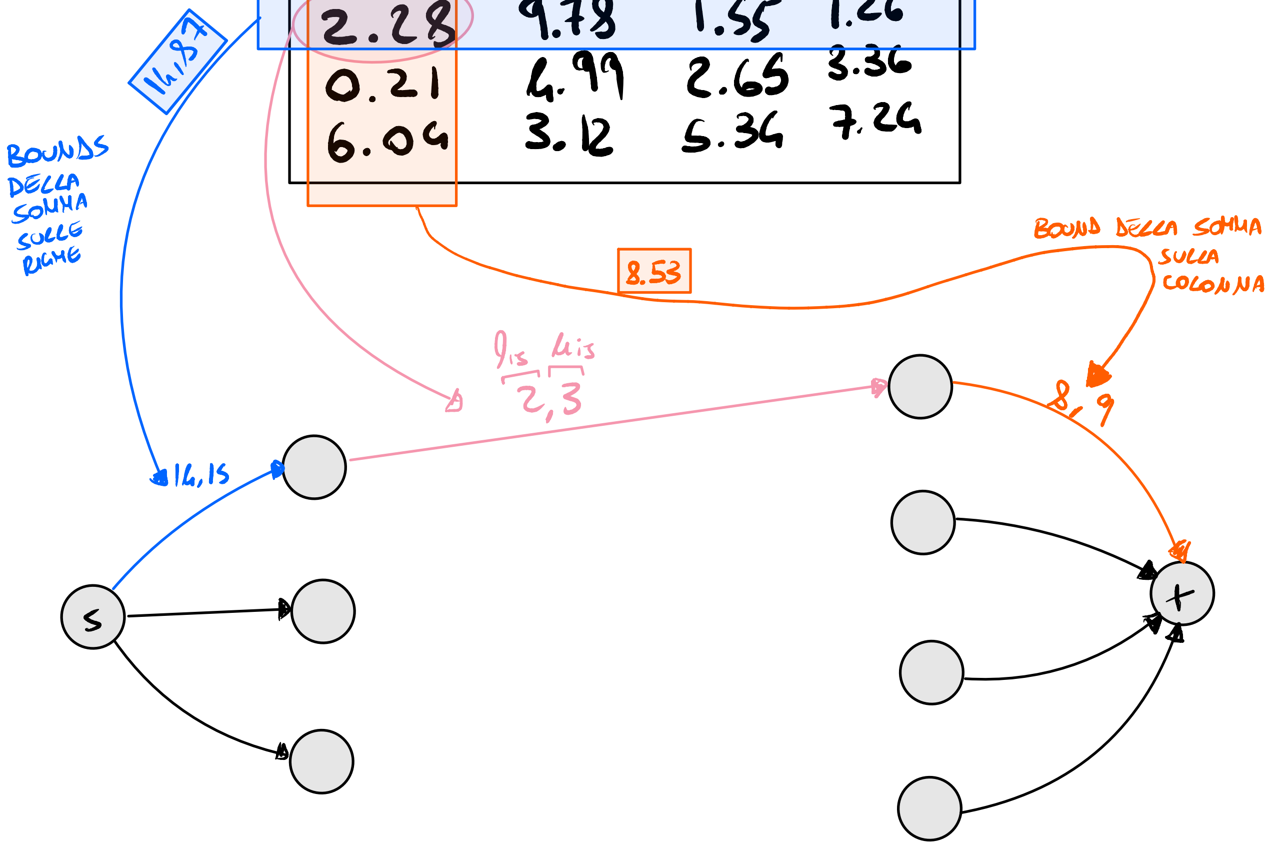
TOPICS PARZIALE

- 1 Ex/Q → FLOW CON LOWER BOUND
  - ↳ MATRIX ROUNDING
- 2 Ex/Q → PREFLOW/PUSH ALGORITHM
- 3 Ex/Q → GLOBAL MIN CUT

MATRIX ROUNDING

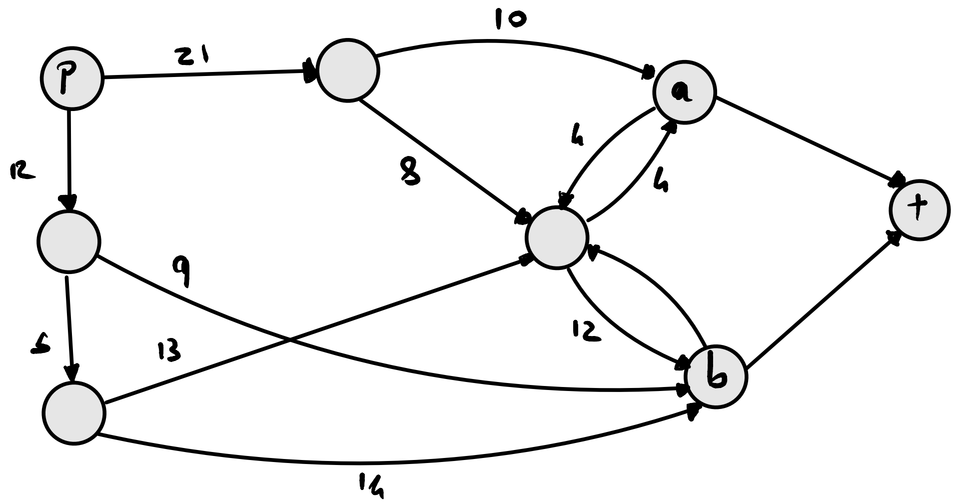
Data la matrice M, trova un "CONSISTENT ROUNDING"

2.28	9.78	1.55	1.26
0.21	4.99	2.65	3.36
6.04	3.12	5.36	7.24



Flow con lower bound

Dato un logistic distribution network spediamo "goods" da p ai nodi a e b, sia k il numero massimo di goods spediti da a, come lo incrementiamo del 10%?

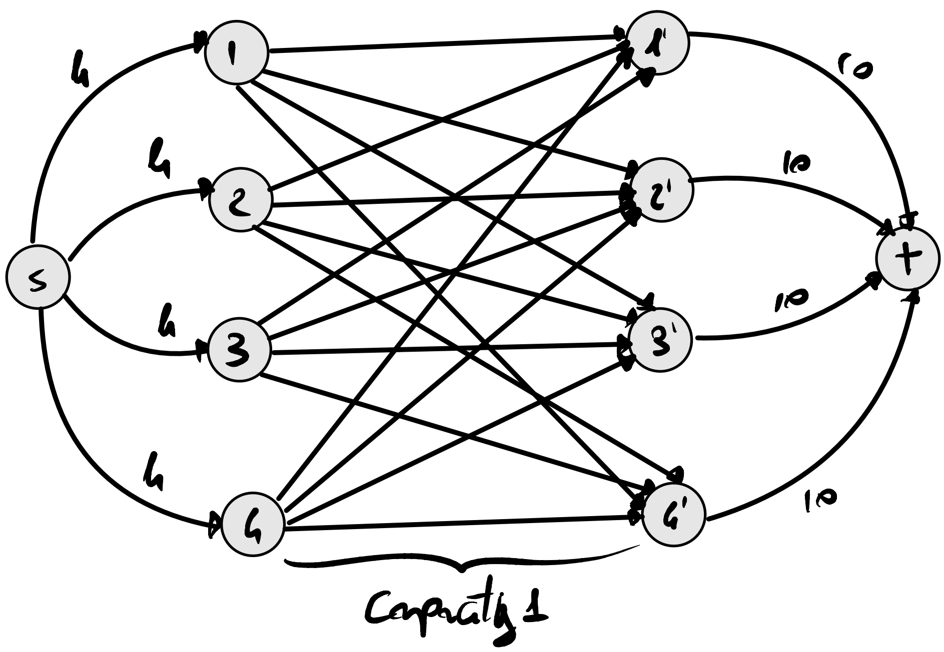


risposta

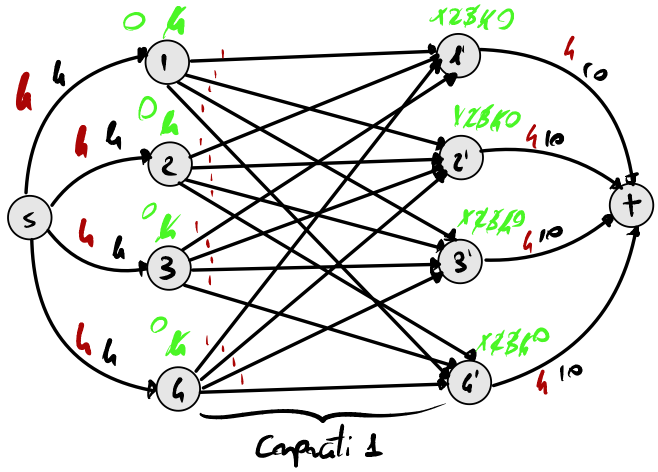
- Dobbiamo fissare il lower bound su (a,t) e fissare la capacity su (b,t) pari al flow nella soluzione ottima
- Incrementare la capacity sul min cut
- Verificare la feasibility
- Risolvere il MAX FLOW

Push label

Trova la sequenza di Push operations sul grafo e confronta con l'FF Algo



· troviamo il preflow e settiamo e e d

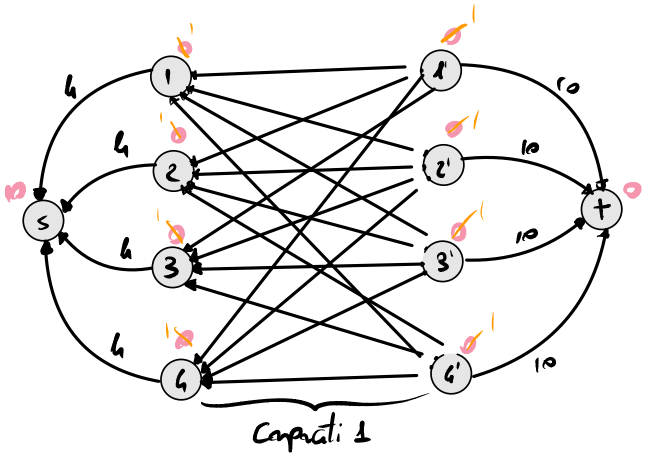


PUSH LABEL ALGORITHM

INIZIALIZZA: Preflow X, Labels d, Eccessi e

WHILE (X NOT FEAS FLOW)

- Scegli NODO ATTIVO i (con e > 0) su G(x)
- WHILE (∃ ARCO AMMISSIBILE (i,j) t.c. $d_i = d_j + 1$)
  - PUSH (i,j)
- IF (i è ancora attivo)
  - RELABEL(i)



· In totale sono $\boxed{16+4+4=24}$

→ 4 Push per ogni nodo a sx

→ 1 Push per ogni nodo a dx

→ 1 Push per ogni nodo a sx per inizializzare il preflow

· FF trova 16 cammini aumentanti

FLOW CON LOWER BOUNDS

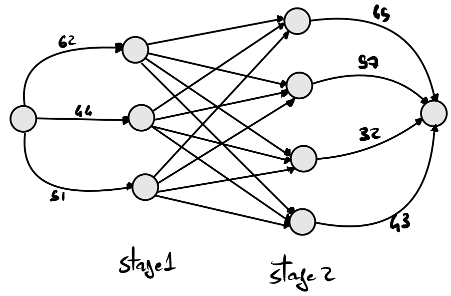
· Dato un manufactoring process fatto da 2 stage consecutivi, ogni stage ha le sue PRODUCTION BAY

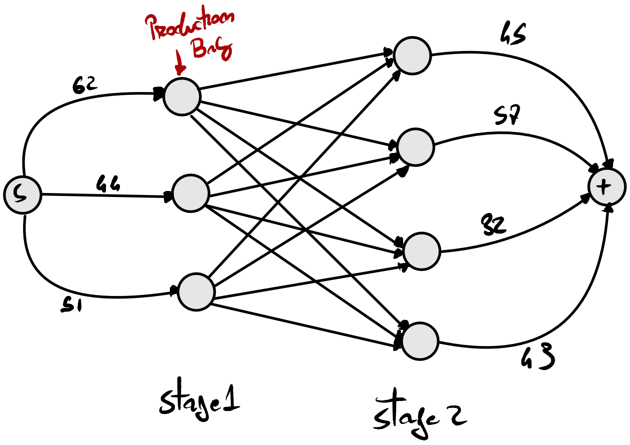
· Allo stage 2, una bay è detta efficiente se lavora al 75% della propria capacity

· Esiste un plan per cui ogni bay dello stage 2 lavora efficientemente?

risposta

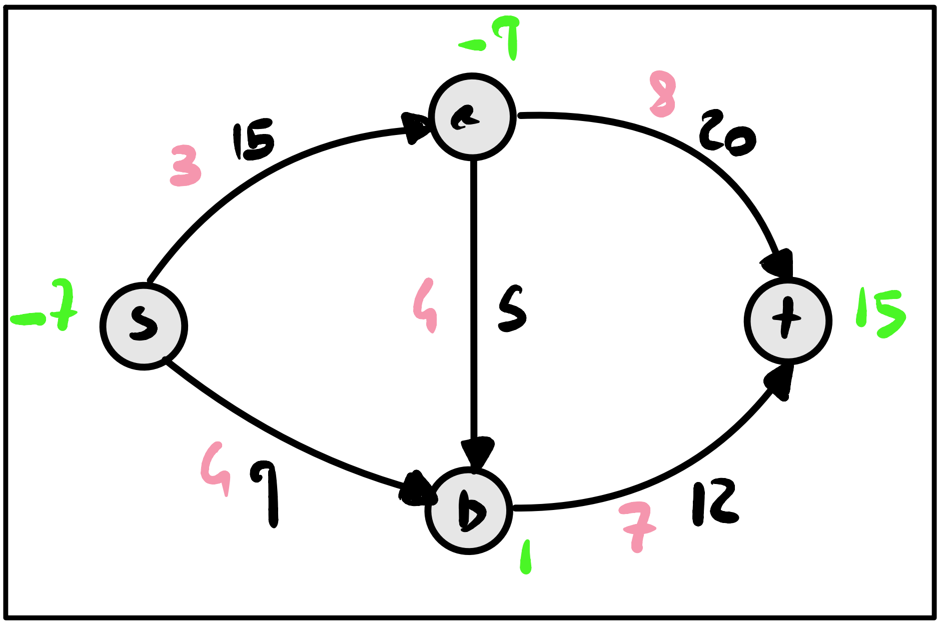
· Imponiamo l_{ij} e check FEASIBILITY



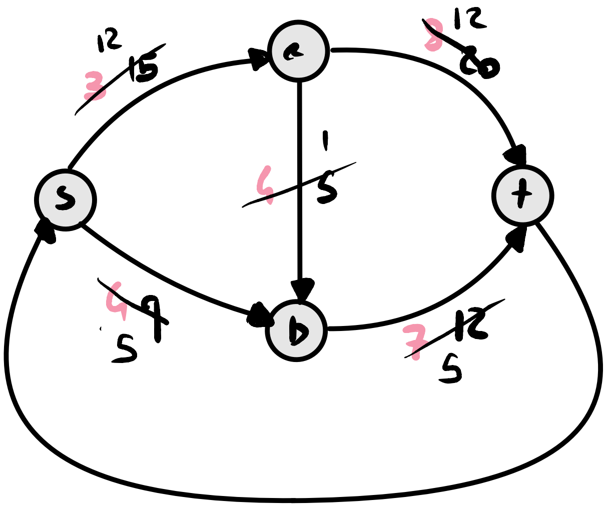


Flow con lower bound

- Eccessi
- Lower bounds



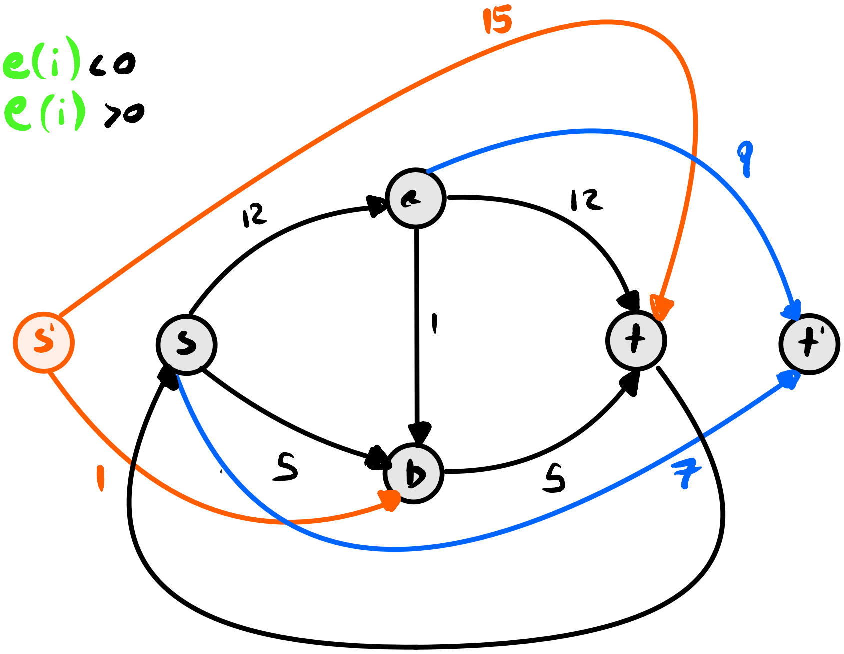
(1) Scaliamo $x'_{ij} = x_{ij} - l_{ij}$
Aggiungiamo (t, s)



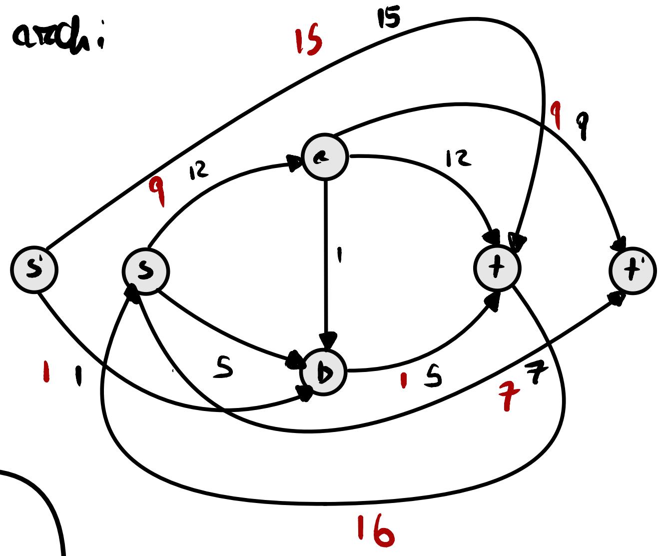
(2) Introduciamo s', t'

Aggiungiamo $\begin{cases} \bullet\ (i, t')\ \forall i \text{ con } e(i) < 0 \\ \bullet\ (s', i)\ \forall i \text{ con } e(i) > 0 \end{cases}$

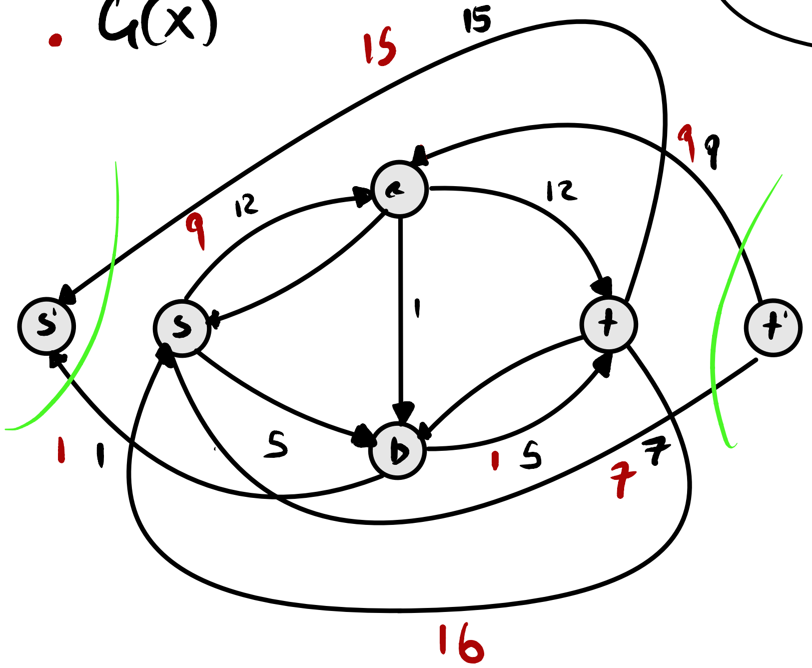
Ogni arco ha capacity $|e(i)|$



(3) FF ⇒ se saturiamo i nuovi archi: esiste FEAS FLOW in G



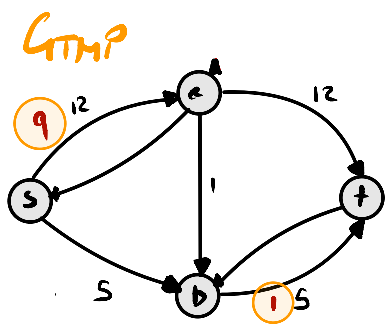
- INIZIALIZZIAMO FLOW
- G(x)



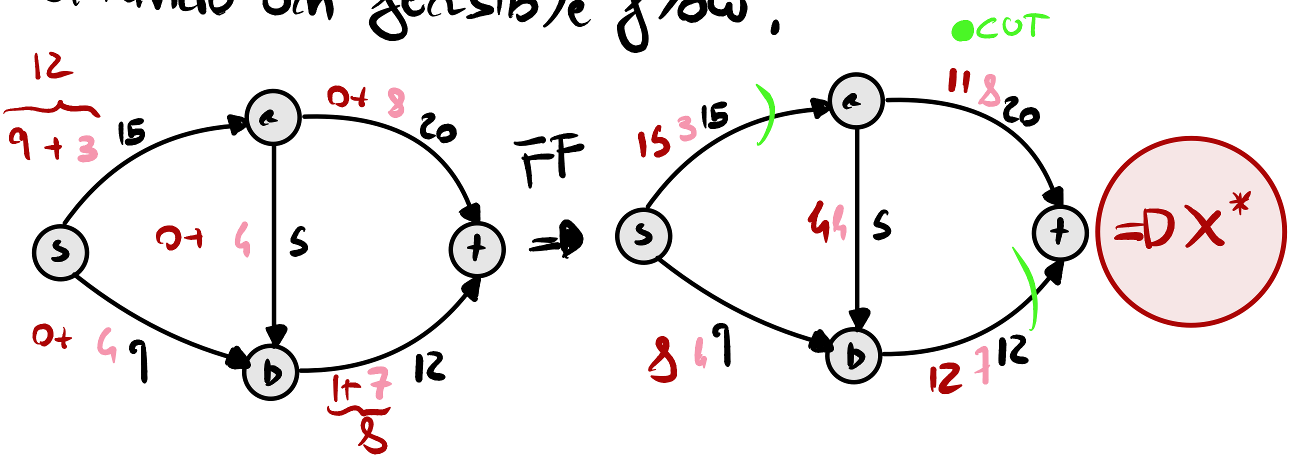
⇒ G DI PARTENZA AMMETTE FEASIBLE FLOW

(4) Rimuoviamo s', t' e (s,t) ottenendo G_{TMP} con X^{TMP} il flow di tale grafo

G_{TMP}

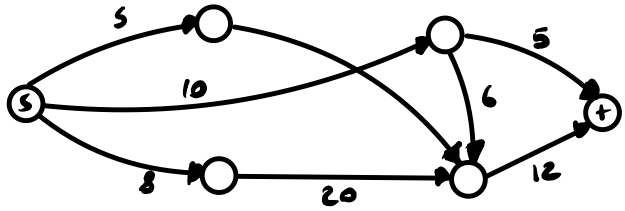


5) Nel grafo di partenza settiamo $X_{ij} = l_{ij} + X^{TMP}_{ij}$ ottenendo un feasible flow!



$= DX^*$

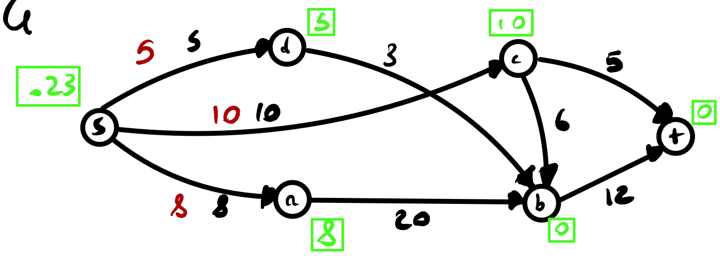
PREFLOW PUSH



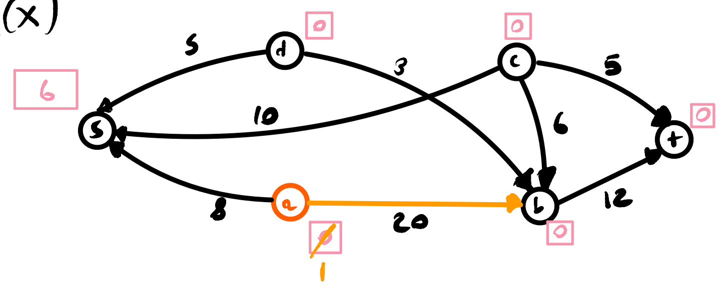
• Initialization

- CAPACITÀ
- FLOW
- ECCESSI
- LABEL
- SELECTED ACTIVE NODE
- PUSH / RELABEL

G

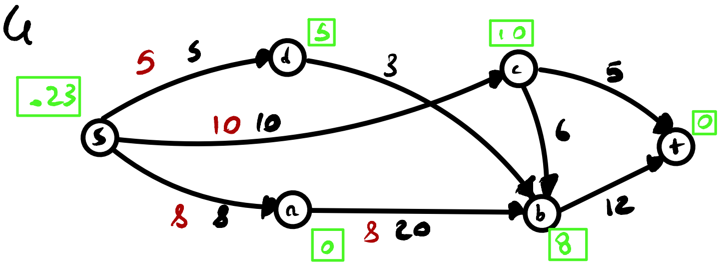


G(x)

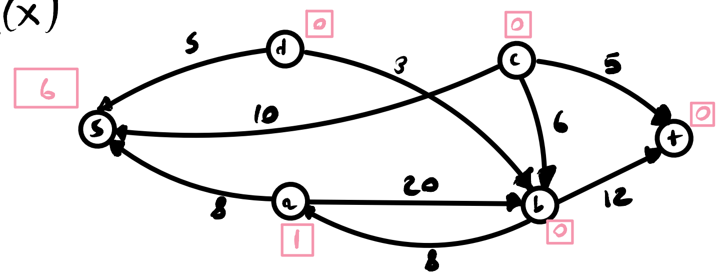


• seleziona il nodo attivo a per azzerare il suo eccesso
• RELABEL (a)
• PUSH ((a,b))

G

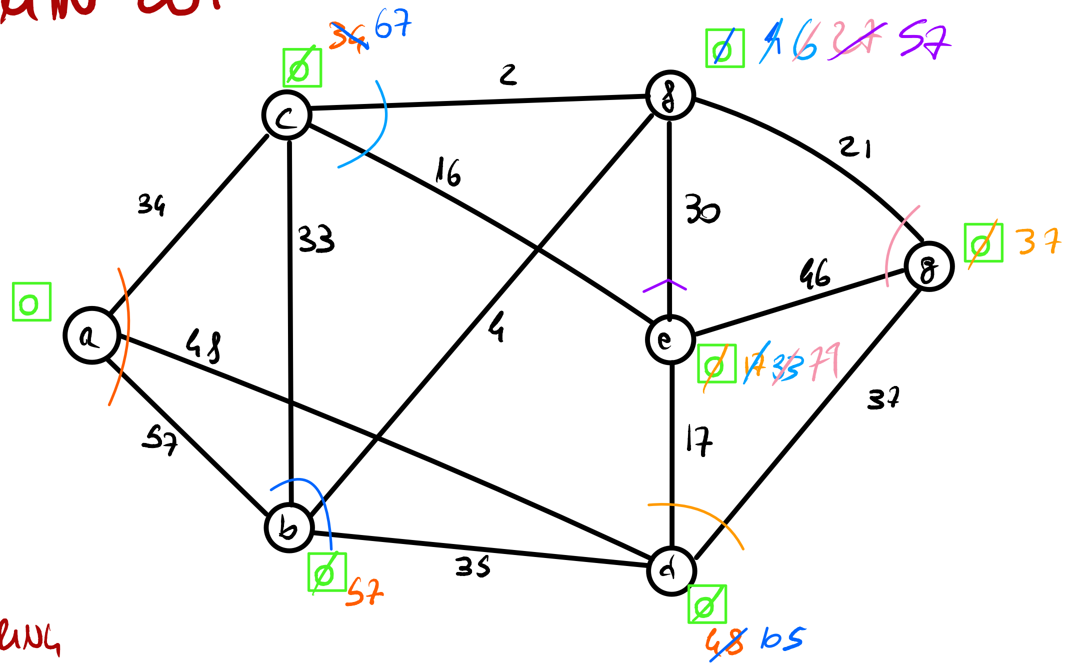


G(x)



...

GLOBAL MIN CUT



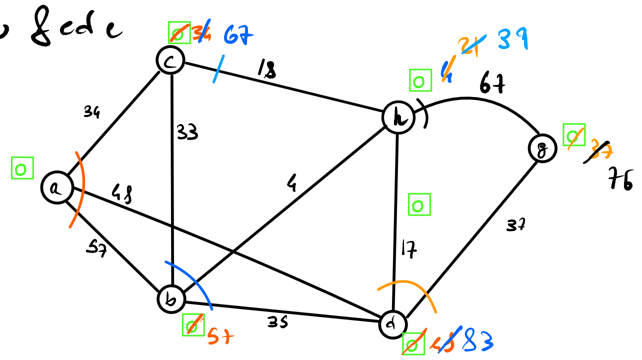
(1) LEGAL ORDERING

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

$$\underbrace{\lambda(G, f, e) = 57}_{= u(\delta(f))}$$ // valore del minimo (e,f)-cut in G

(2) NODE IDENTIFICATION

• su f e d e



(1) $O = \{a, b, d, c, h, g\}$

$\lambda(G, h, g) = 76$

(2) ...

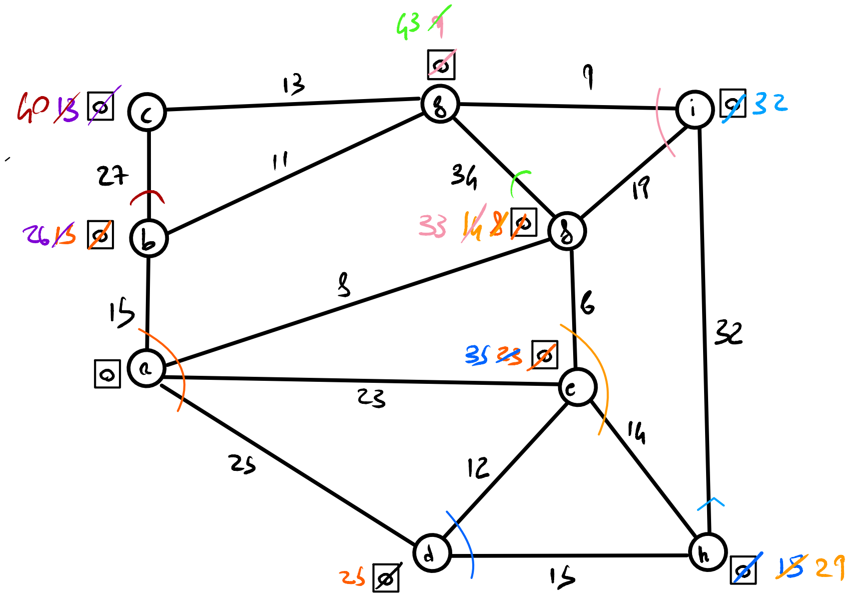
(1) Troviamo Legal ordering

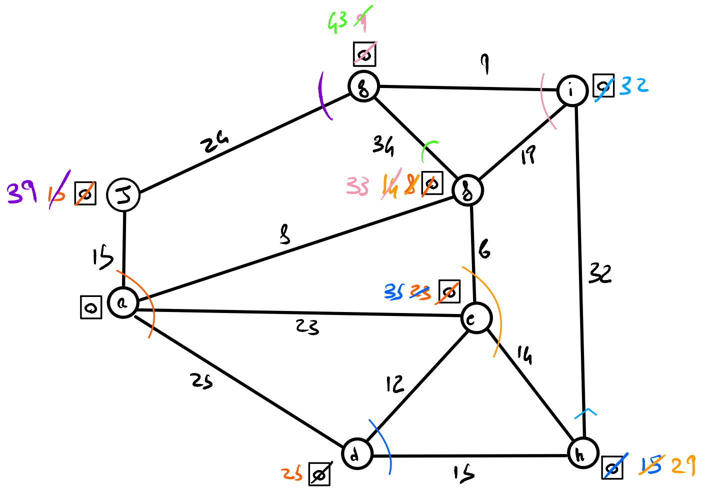
$O = \{a, d, e, h, i, f, g, b, c\}$

(2) Troviamo il valore del minimo (c,b)-cut

$\lambda(G, c, b) = d(\delta(c)) = 40$

(3) Identifichiamo i nodi c e b

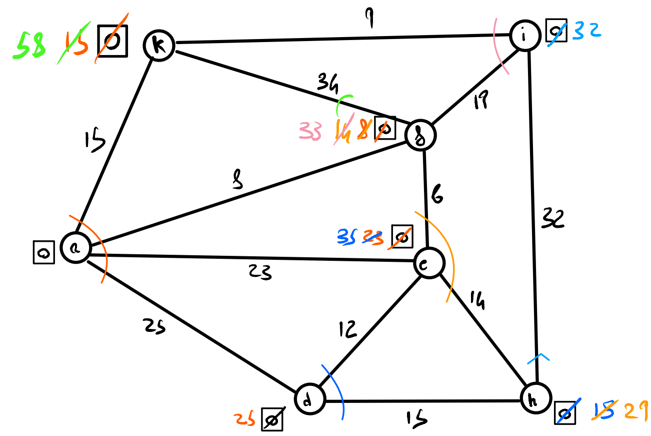




(1) $O = \{a, d, c, h, i, f, g, J\}$

(2) $\lambda(G, J, g) = 39$

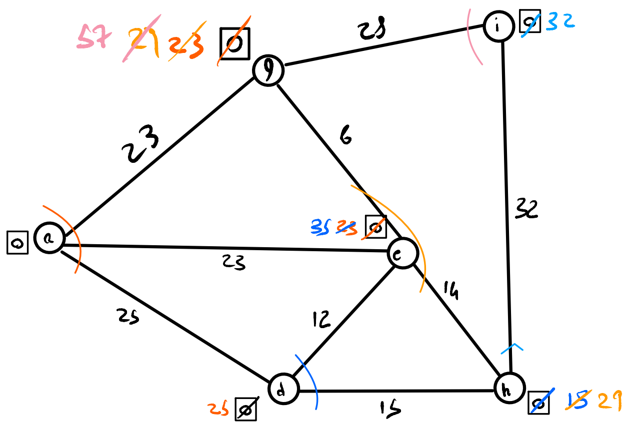
(3) IDENTIFICATION (J, g)



(1) $O = \{a, d, c, h, i, f, k)$

(2) $\lambda(G, k, f) = 58$

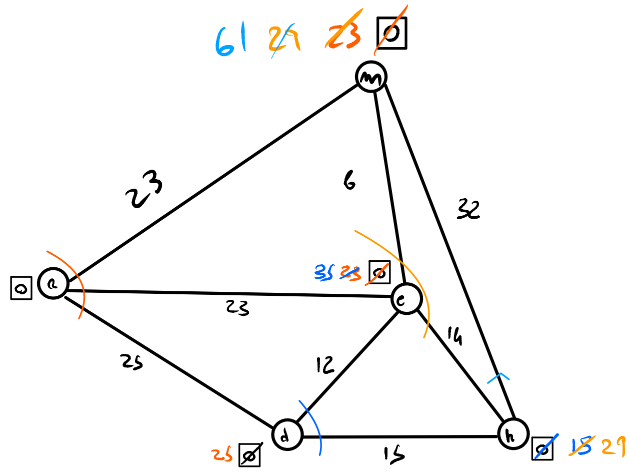
(3) IDENTIFICATION (k, f)



(1) $O = \{a, d, c, h, i, 9\}$

(2) $\lambda(G, 9, i) = 57$

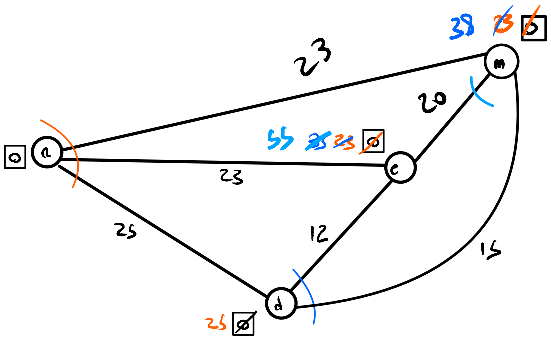
(3) IDENTIFICATION (9, i)

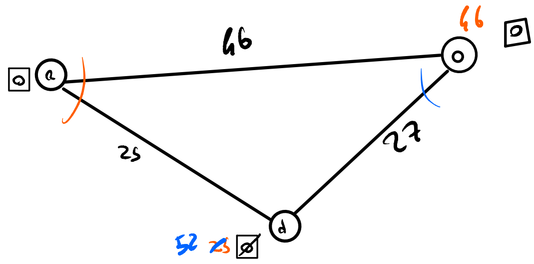


(1) $O = \{a, d, c, h, m)$

(2) $\lambda(G, m, h) = 61$

(3) IDENTIFICATION (h, m)





(1) $O = \{a, dm, c\}$

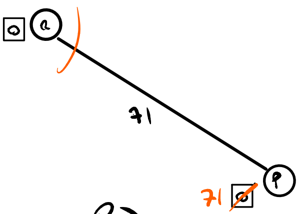
(2) $\lambda(G, e, m) = 55$

(3) IDENTIFICATION (m, e)

(1) $O = \{a, e, d\}$

(2) $\lambda(G, d, o) = 55$

(3) IDENTIFICATION (o, d)

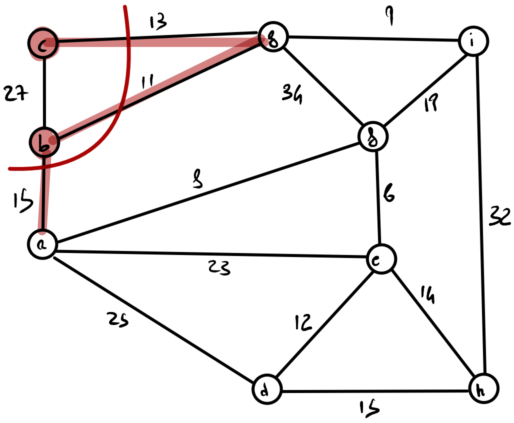


(1) $O = \{a, p\}$

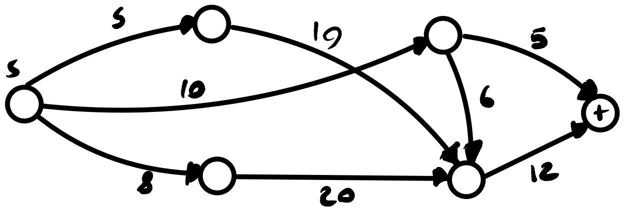
(2) $\lambda(G, p, a) = 71$

(3) IDENTIFICATION (a, p)

GLOBAL MIN CUT = 39



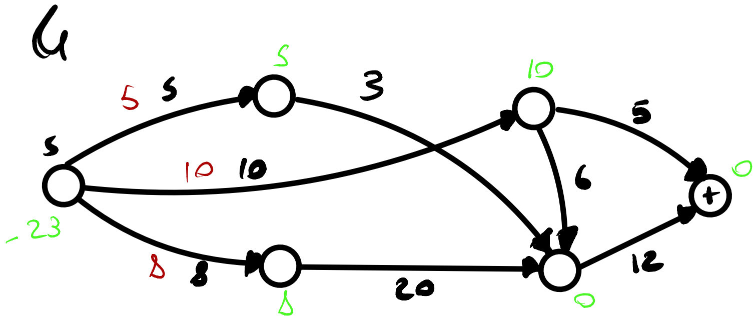
PUSH LABEL ALGORITHM

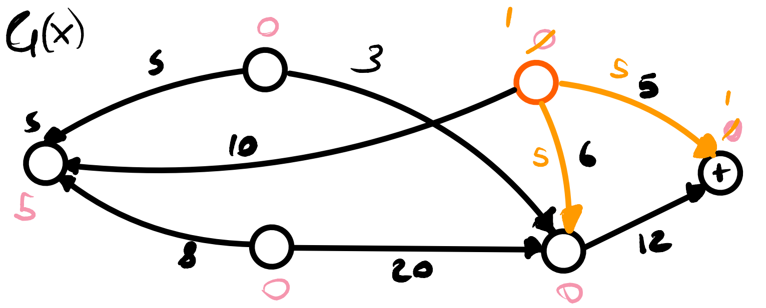


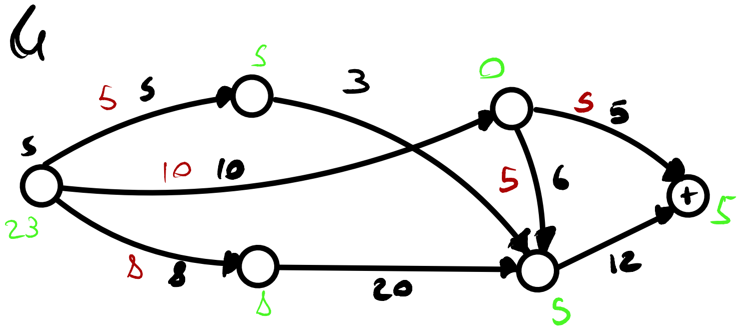
INIZIALIZZA: Preflow X, Labels d, Eccessi e

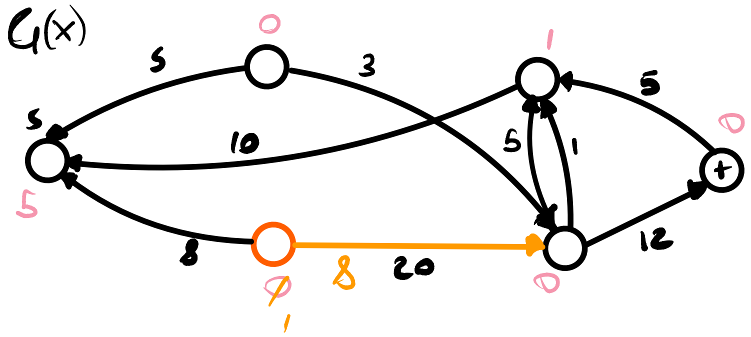
WHILE (X NOT FEAS FLOW)

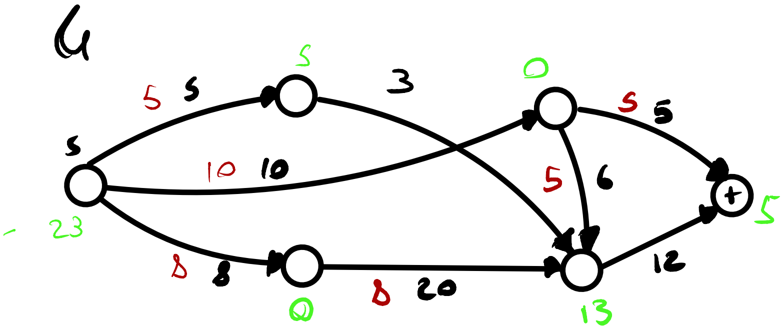
- Scegli NODO ATTIVO i (con e>0) su G(x)
- WHILE (∃ ARCO AMMISSIBILE ((i,j) t.c. $d_i = d_j + 1$))
  - PUSH (i,j)
- IF (i è ancora attivo)
  - RELABEL(i)

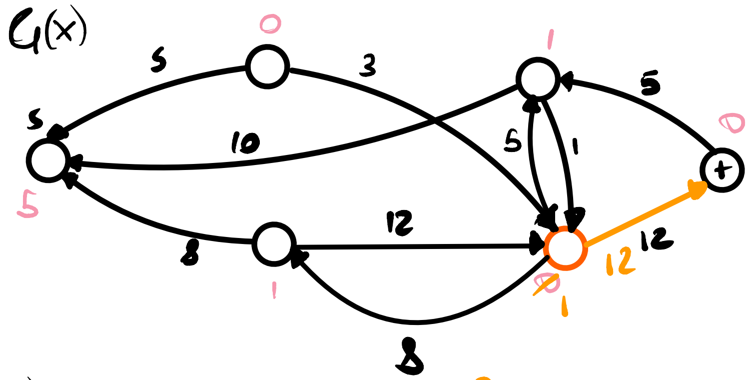


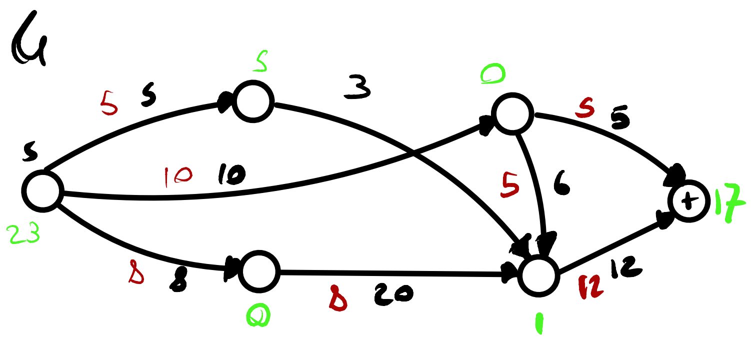


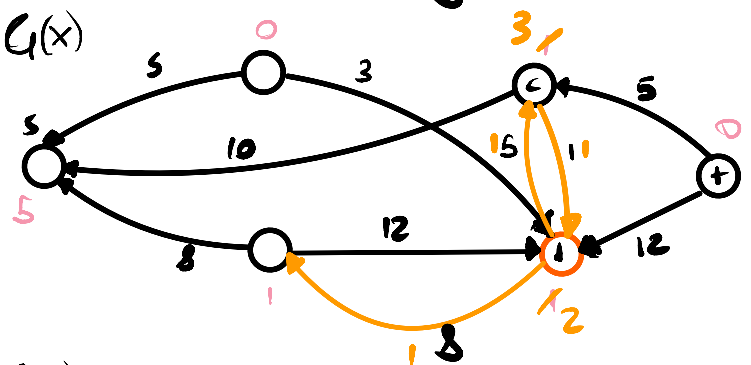


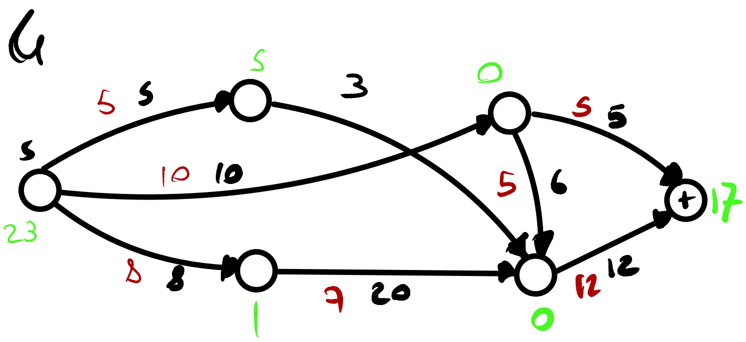


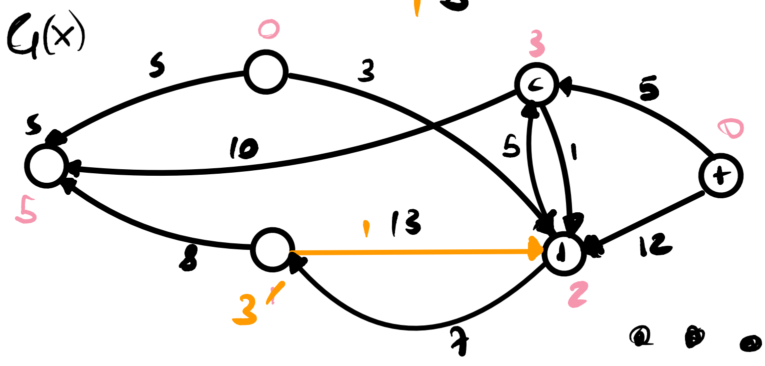






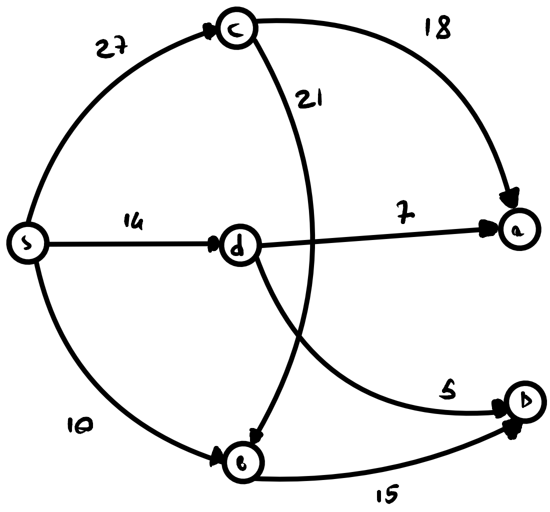




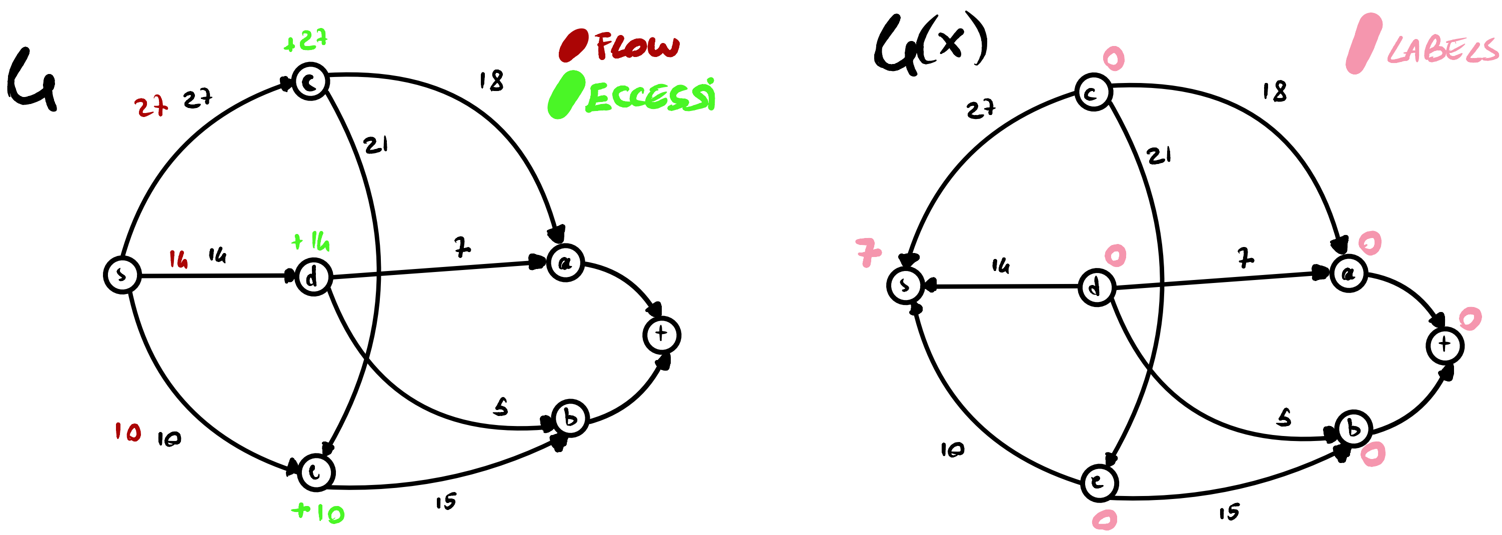


PREFLOW PUSH

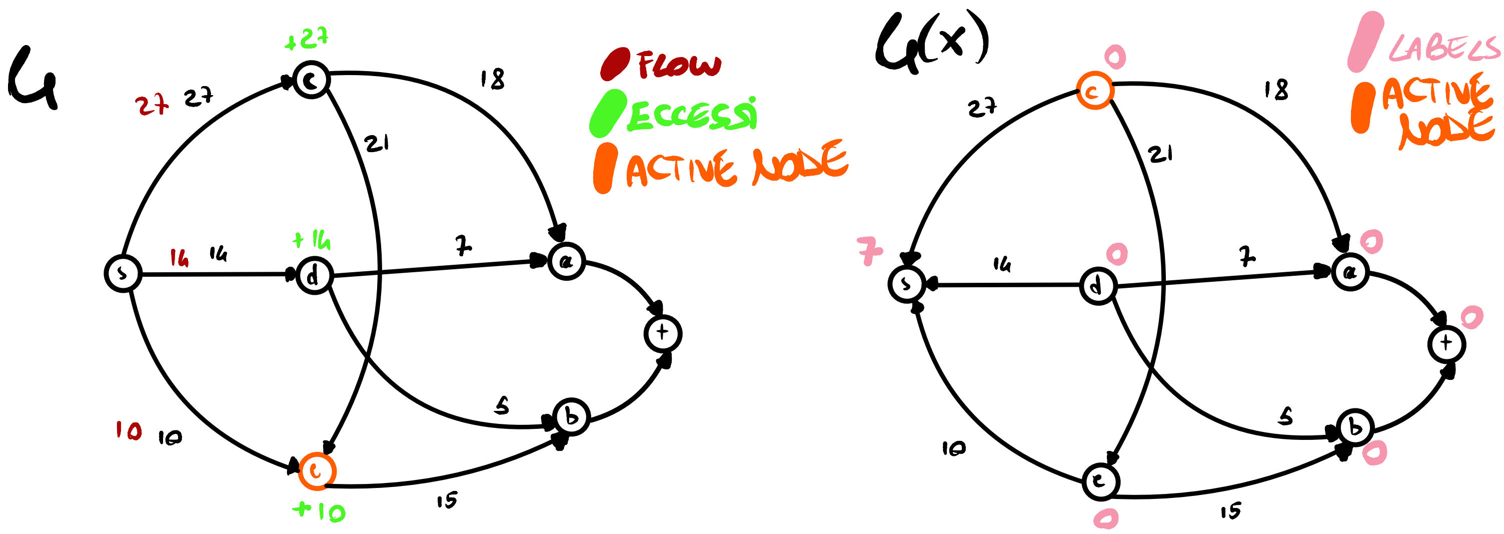
. TROVA IL MAX FLOW s-t TRAMITE a,b CON PREFLOW PUSH



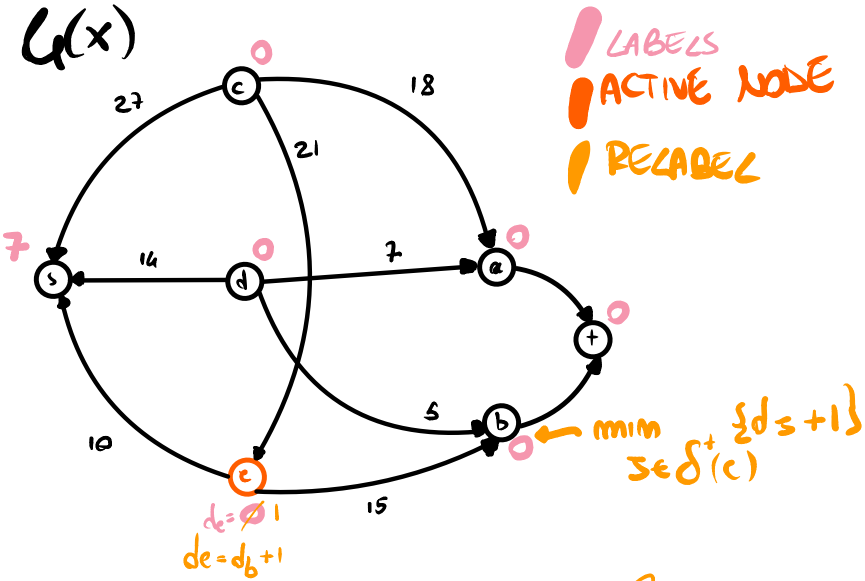
• INITIALIZATION



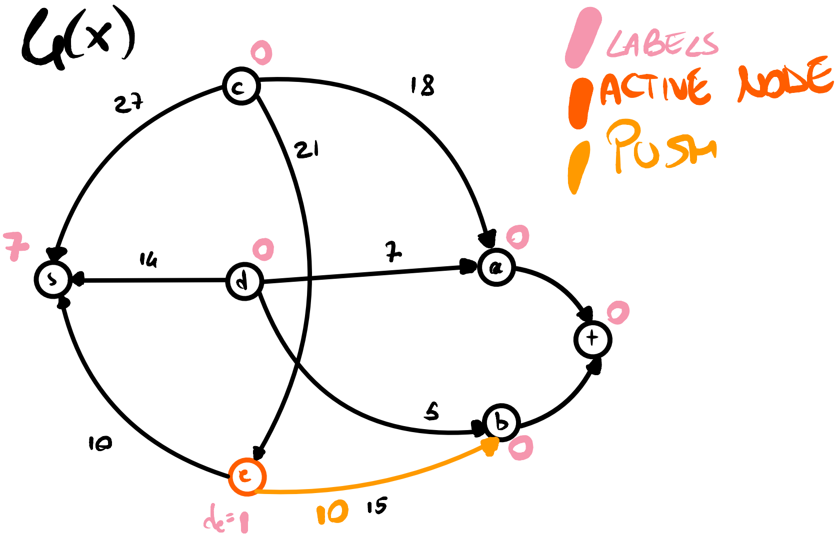
• SELECTION OF AN ACTIVE NODE → {e}



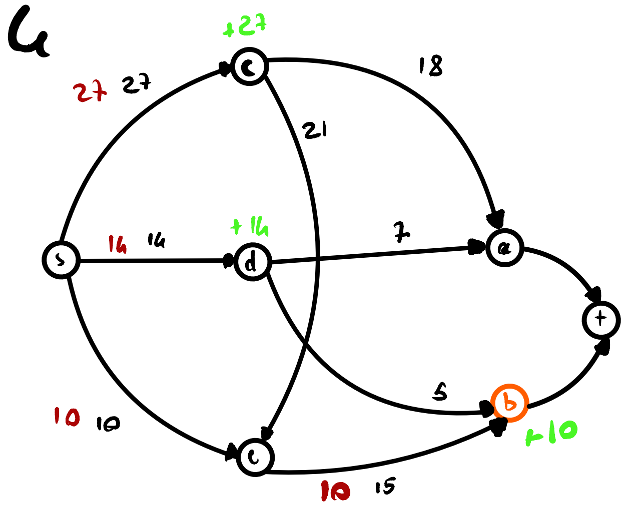
• ∃ ADMISSIBLE ARC $((i,j) \mid d_i = d_j+1)$? NO RELABEL → $d_i = \min_{j \in \delta^+(i)} \{d_j\} + 1$

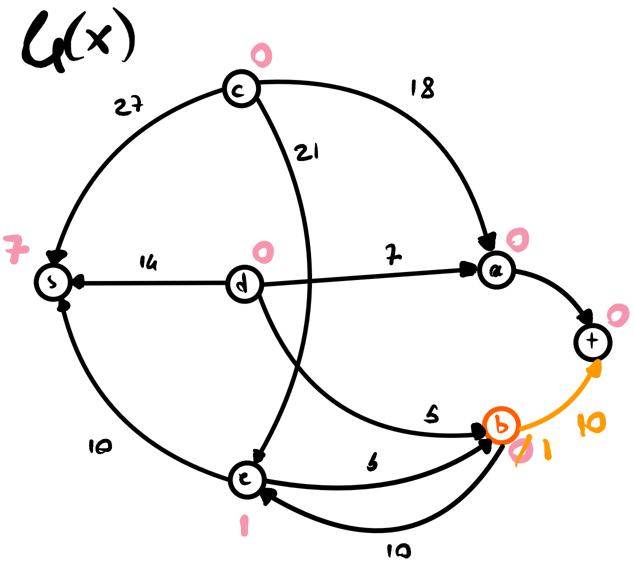


• ∃ ADMISSIBLE ARC $((i,j) \mid d_i = d_j+1)$? SI (e,b) ⇒ PUSH

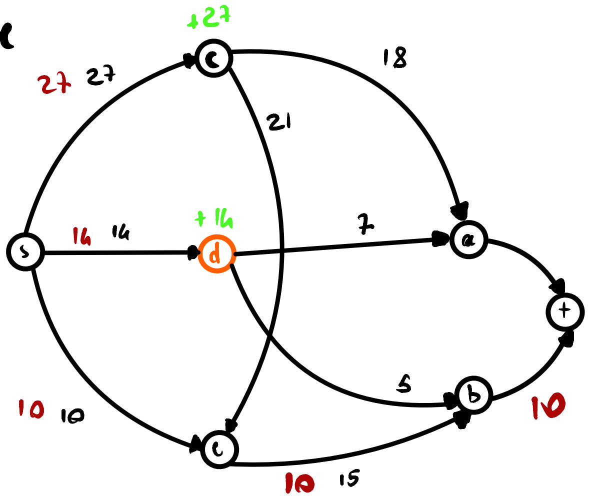


• • •

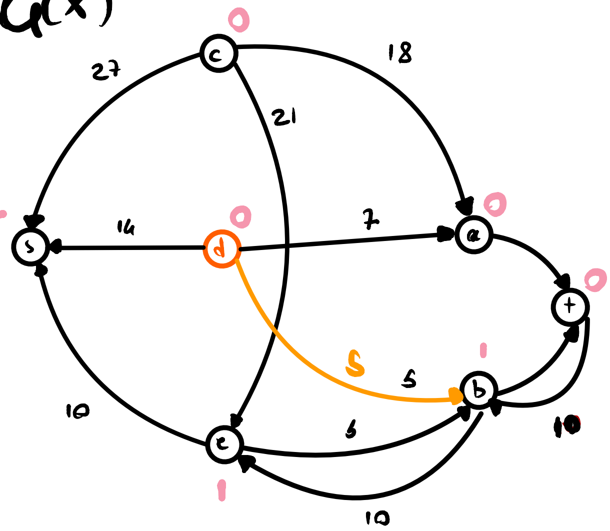




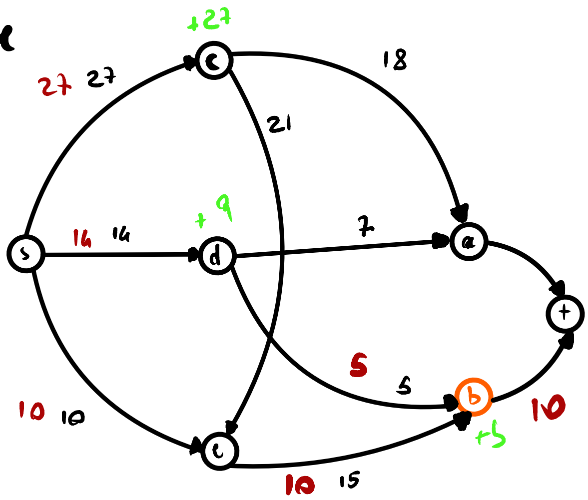
G



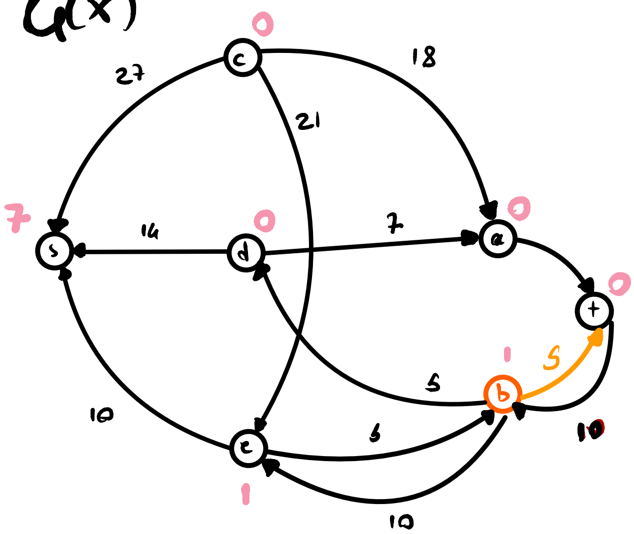
G(x)



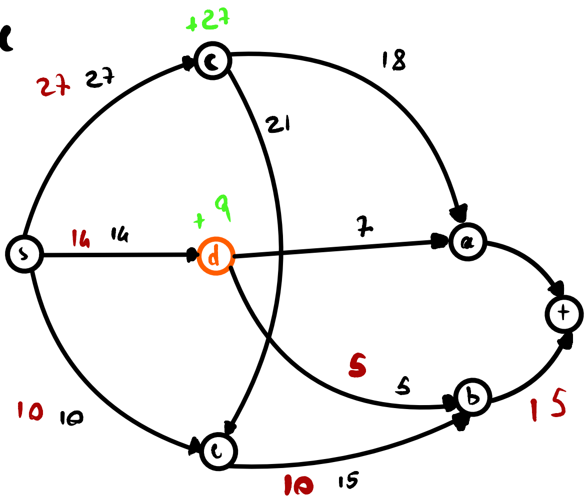
G



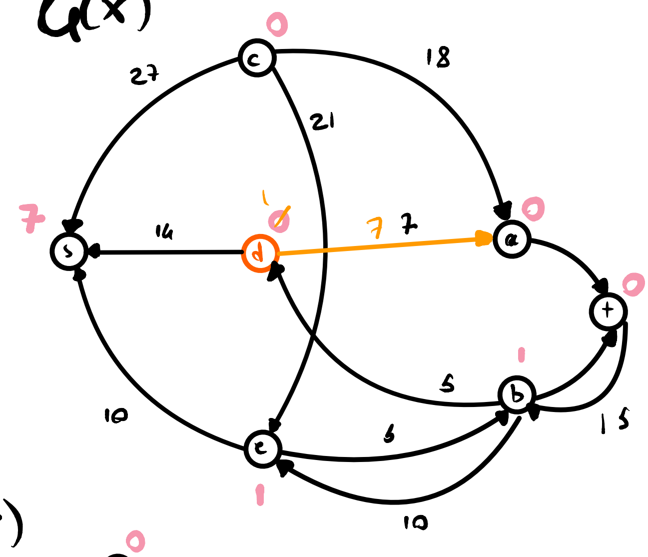
G(x)



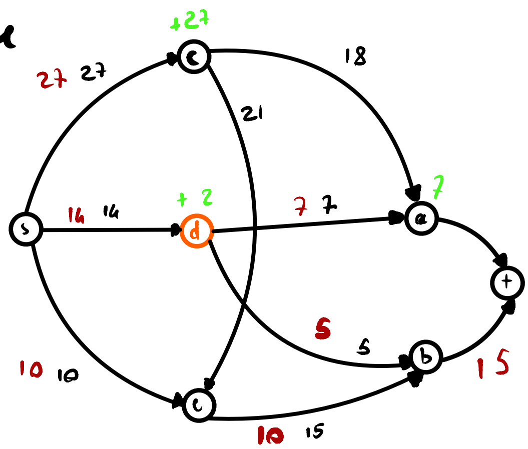
G



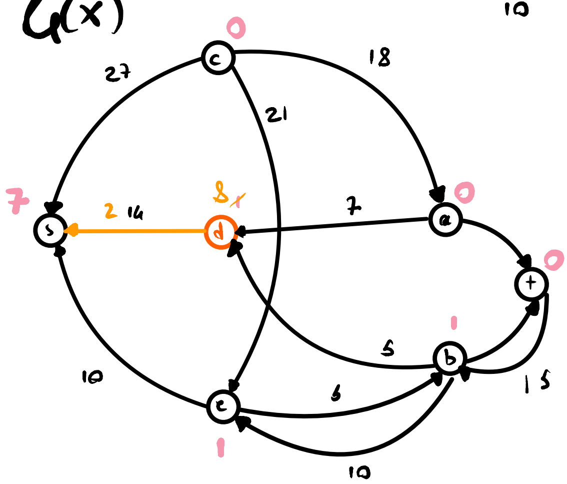
G(x)



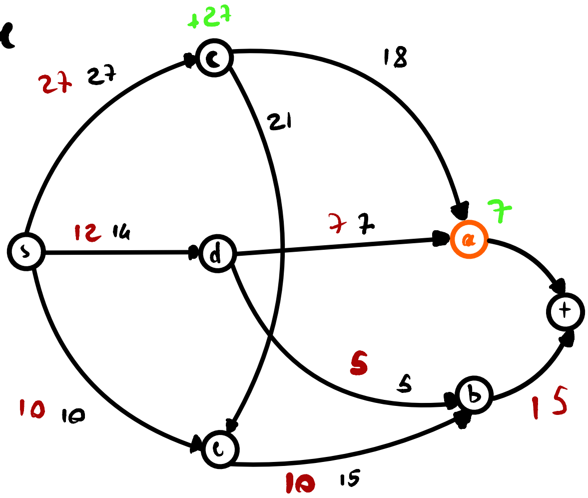
G



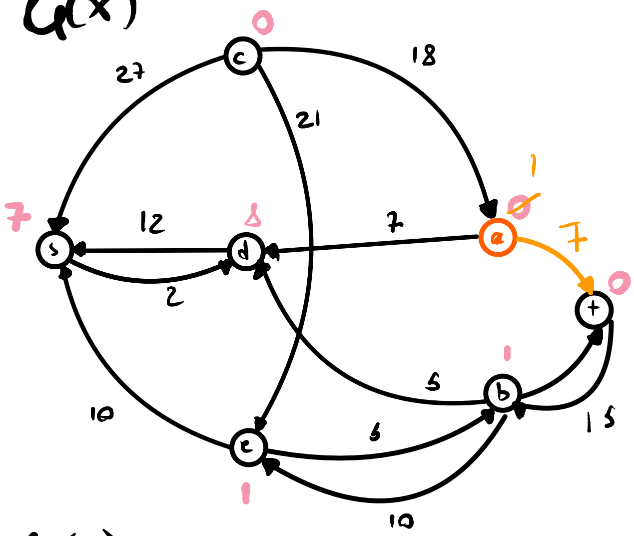
G(x)



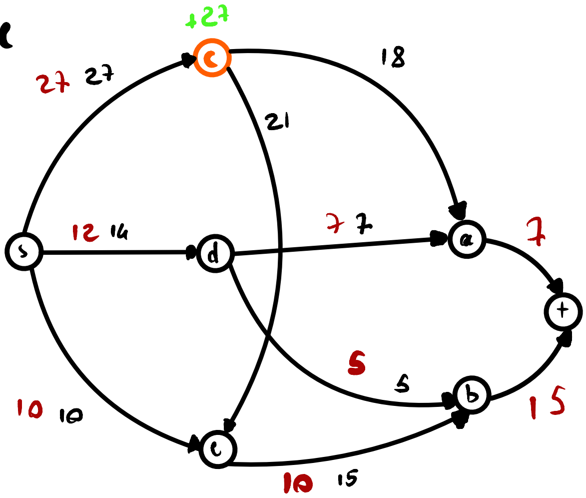
G



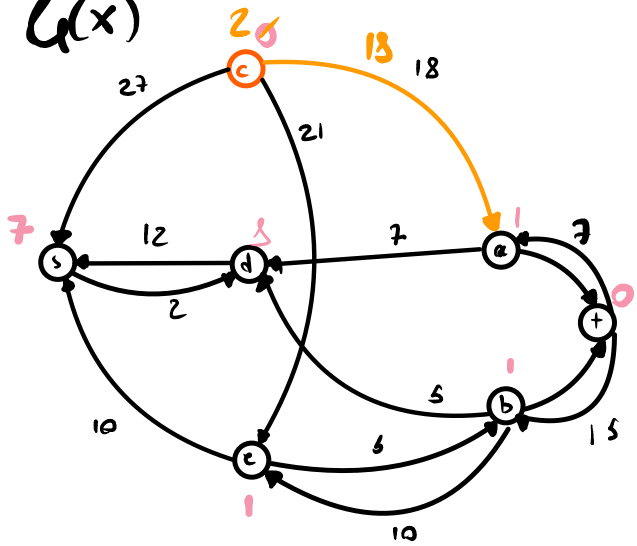
G(x)



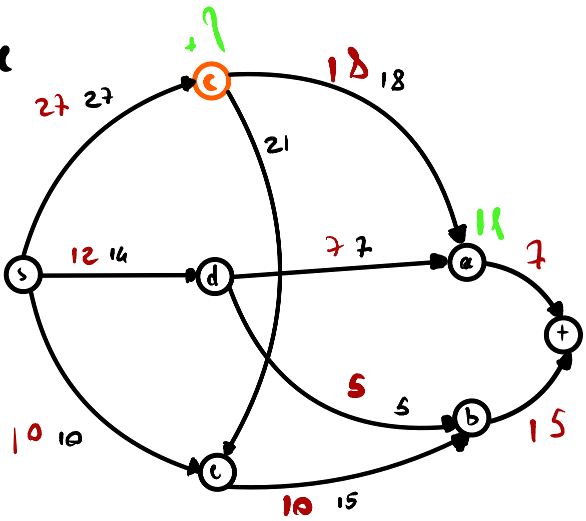
G



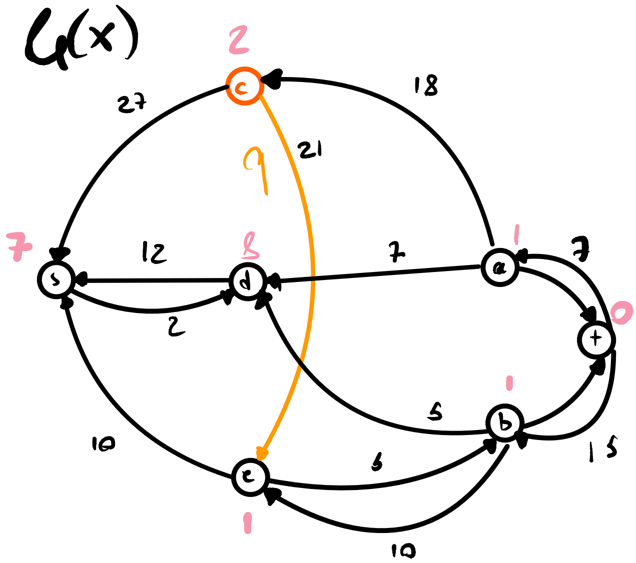
G(x)



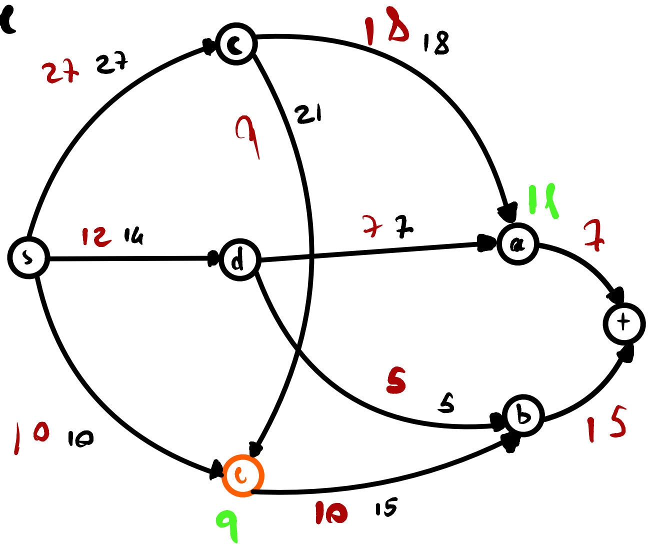
G



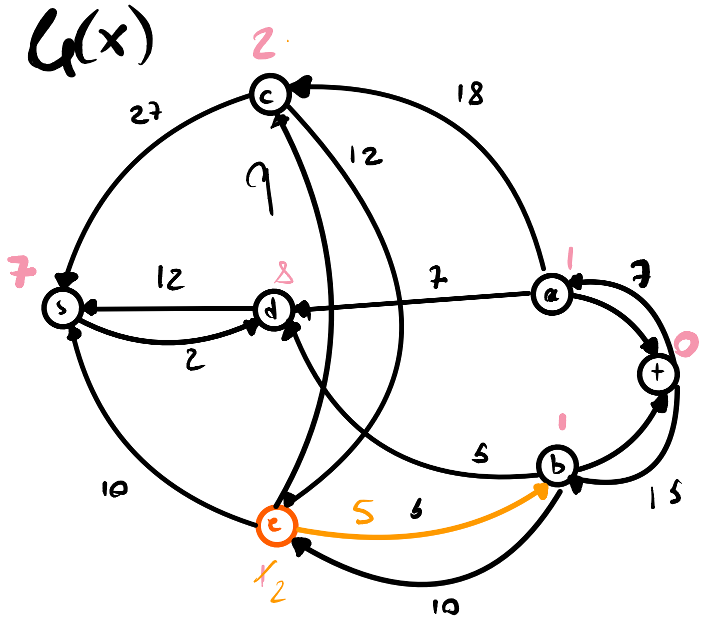
G(x)



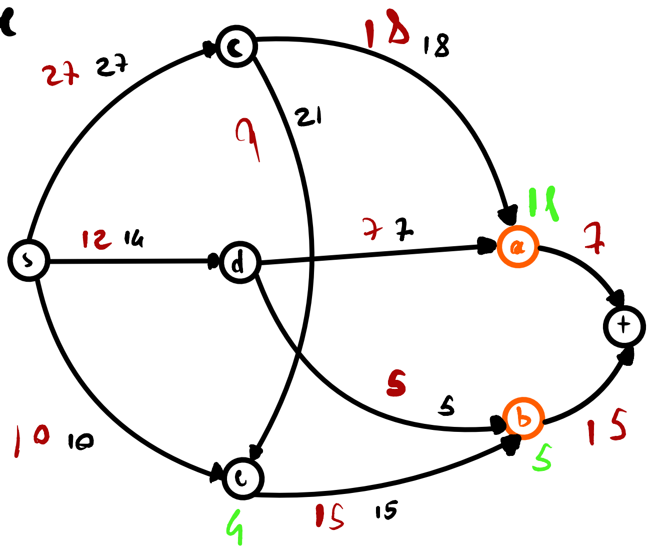
G



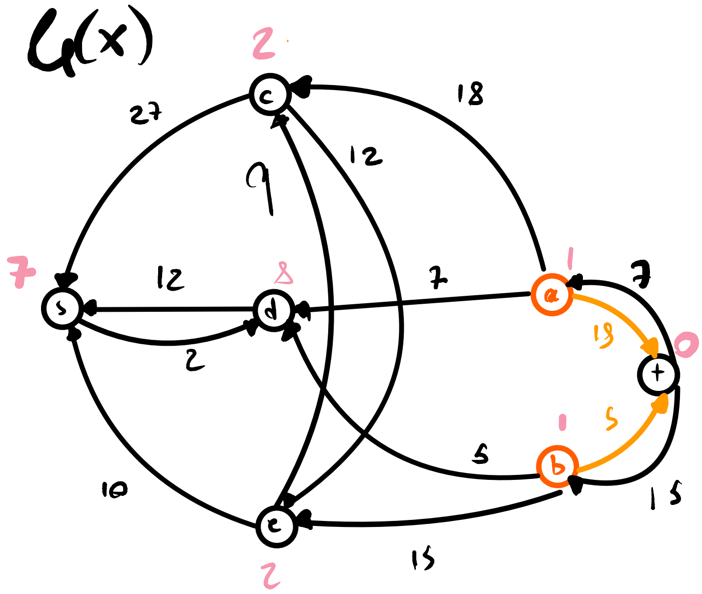
G(x)



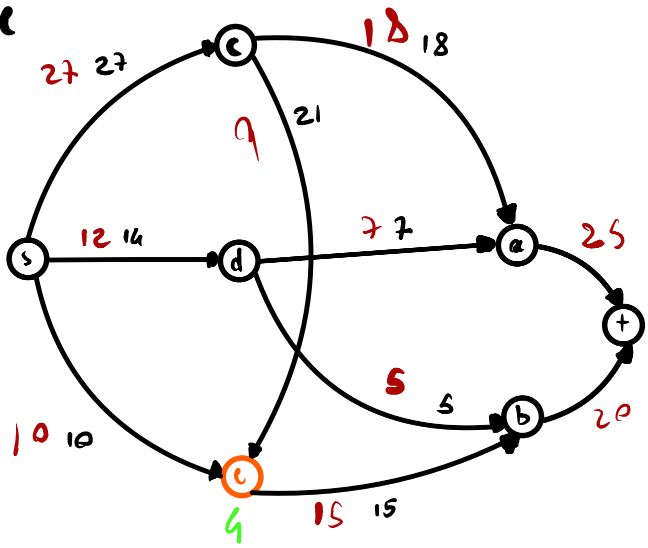
G



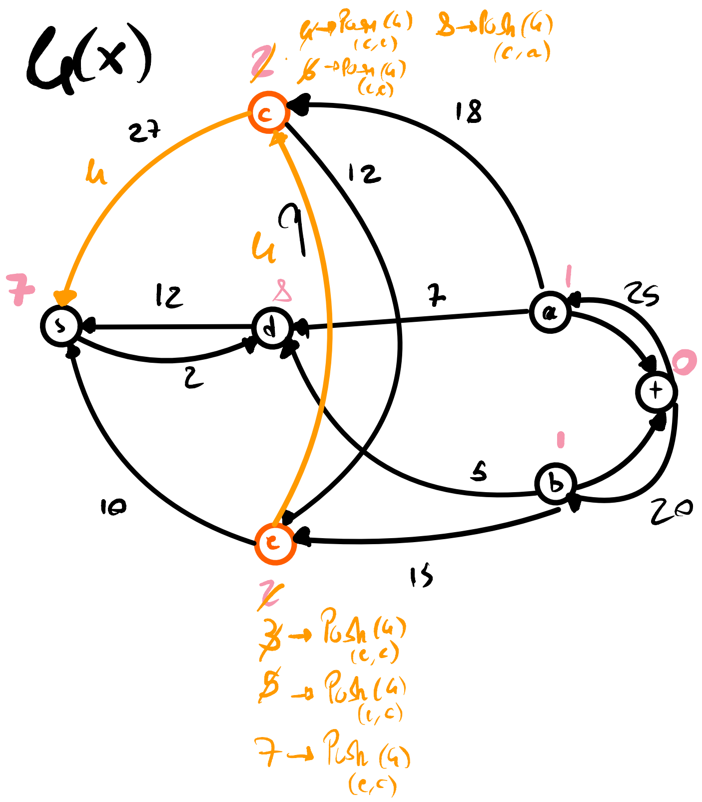
G(x)

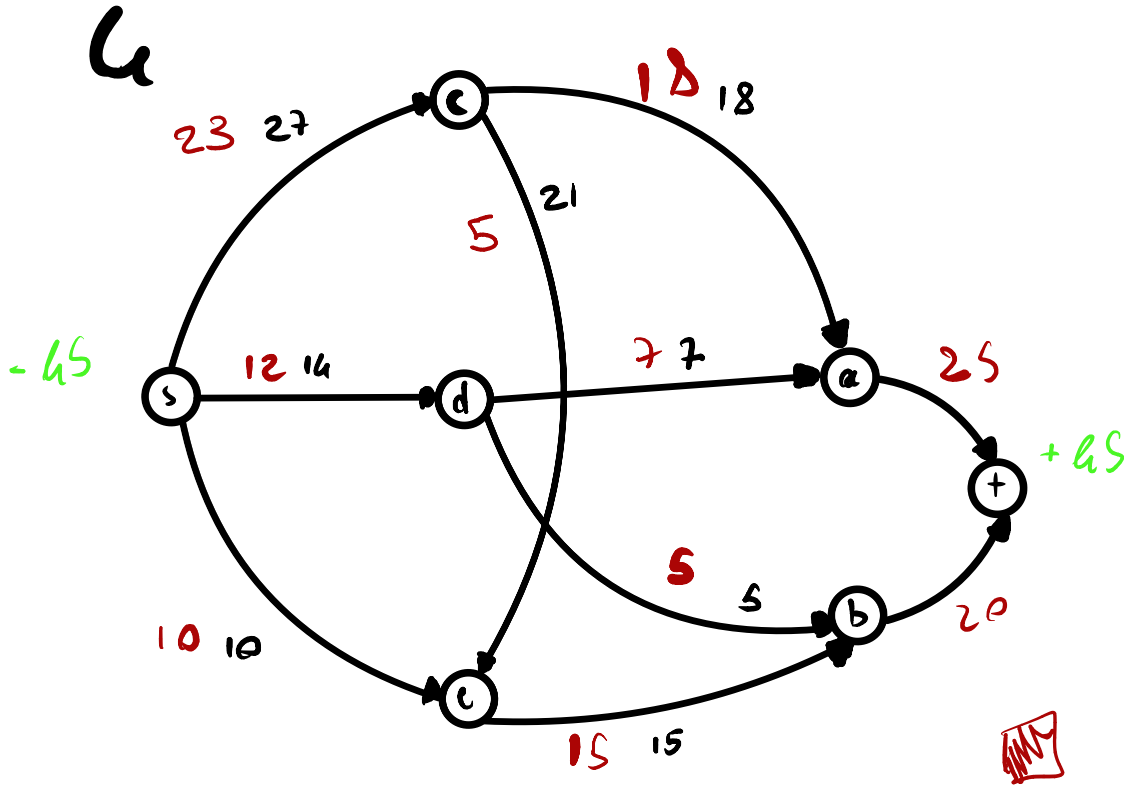


G



G(x)

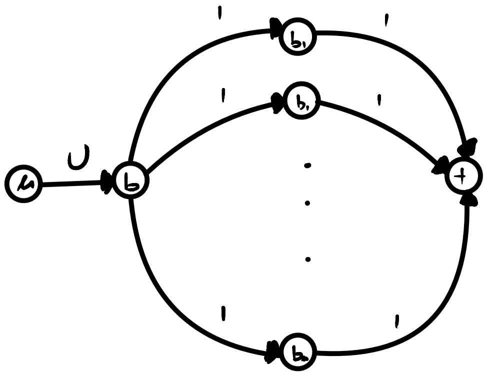



G
s
c
d
e
a
b
t
-45
+45
23 27
18 18
5 21
12 14
7 7
25
5 5
20
10 10
15 15

Question 1

Draw a graph $G=(N,A)$ in which the number of augmentation steps performed by the Ford and Fulkerson algorithm is exactly U, where $U = \max\{u_{ij}\}$ for $(i,j) \in A$

The push-relabel algorithm has a better performance on such a graph?



• FF TROVA U CAMMINI AUMENTANTI

PUSH LABEL ALGORITHM

INIZIALIZZA: Preflow x, Labels d, Eccessi e

WHILE (x NOT FEAS FLOW)

Scegli NODO ATTIVO i (con $e>0$) su $G(x)$

WHILE (∃ ARCO AMMISSIBILE (i,j) t.c. $d_i = d_j + 1$)

PUSH(i,j)

IF (i è ancora attivo)

RELABEL(i)

• PUSH LABEL $1+U+U$

→ 1 per l'inizializzazione

→ U al primo step

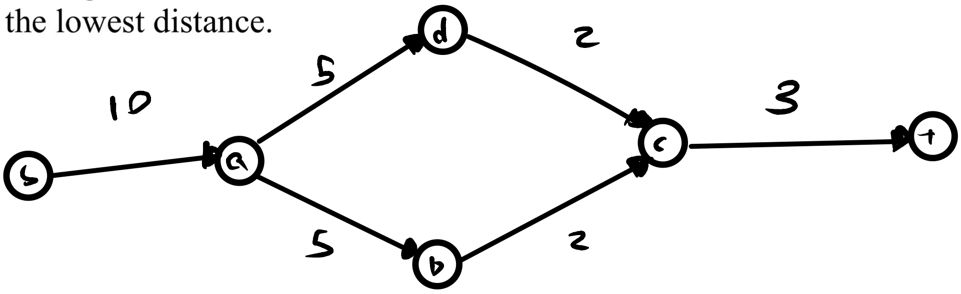
→ U al secondo step

Question 2

In the push-relabel one can define a rule to choose the next active node being processed. Compare on the following graph the two rules:

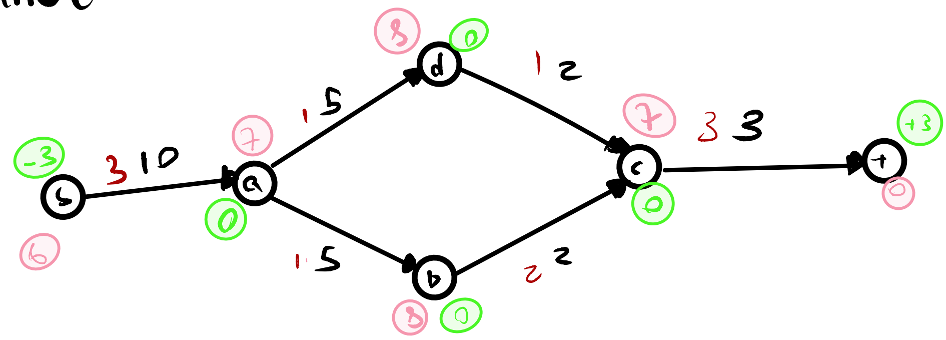
1. Choose the active node with the highest distance.
2. Choose the active node with the lowest distance.

(In both rules break ties arbitrarily)



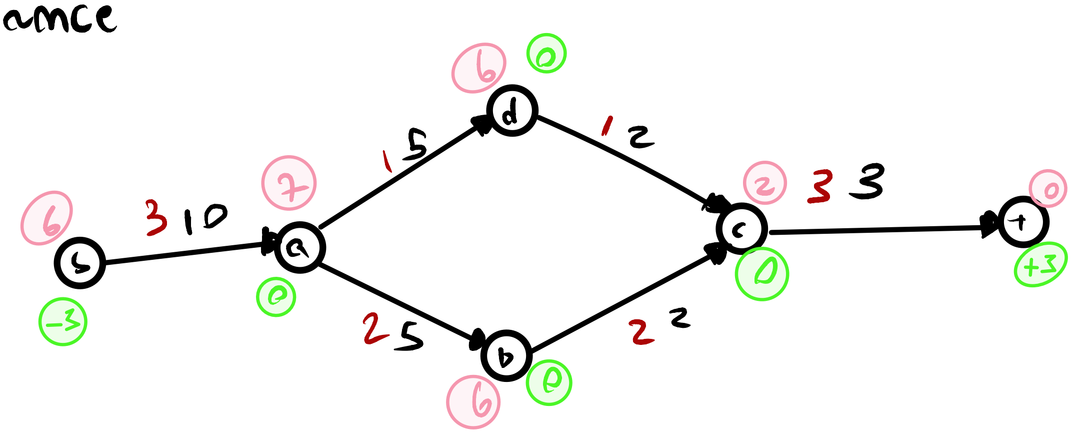
(1) MAX distance

#PUSH → 17



(2) Min distance

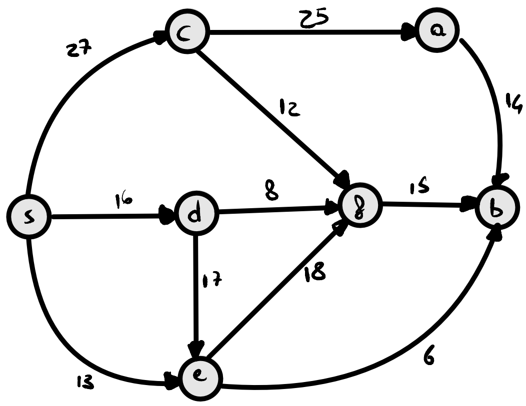
#PUSH → 16



Exercise 1

The following graph G=(N,A) represents a logistic distribution network. Node s is a manufacturing plant, origin of the goods and nodes a and b represent warehouses.

1. Evaluate the maximum quantity of goods that can be shipped from the plant to the warehouses by the preflow-push algorithm



PUSH LABEL ALGORITHM

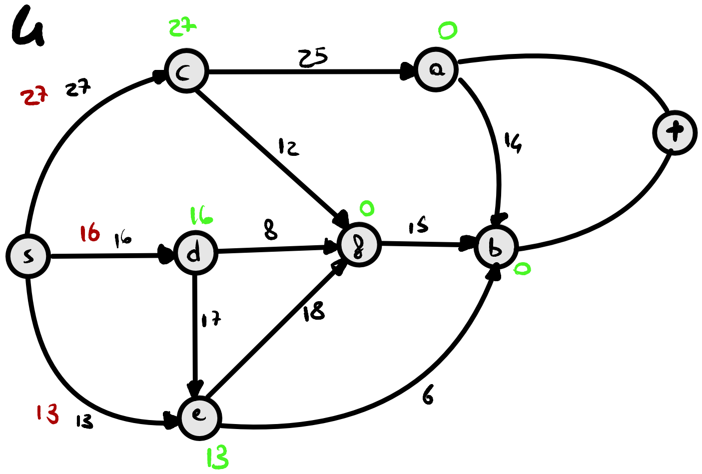
INIZIALIZZA: Preflow X, Labels d, Eccessi e

WHILE (X NOT FEAS FLOW)

- Scegli NODO ATTIVO i (con e > 0) su G(x)
- WHILE (∃ ARCO AMMISSIBILE (i,j) t.c. $d_i = d_j + 1$)
  - PUSH (i,j)
- IF (i è ancora attivo)
  - RELABEL(i)

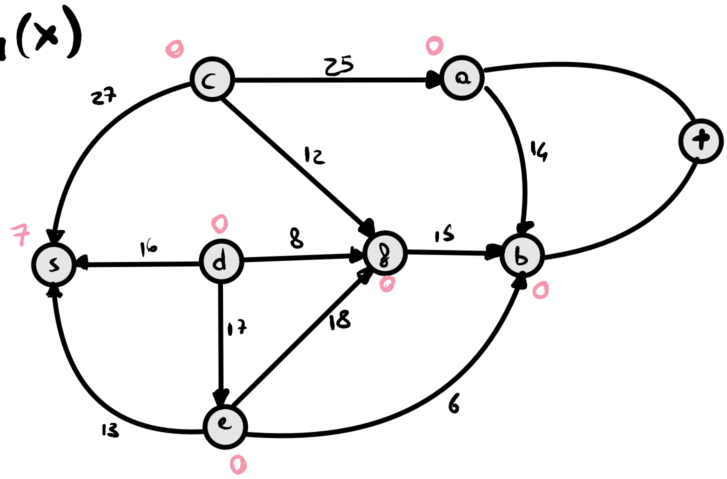
. Inizializzo Preflow x e gli Eccessi e su G

G



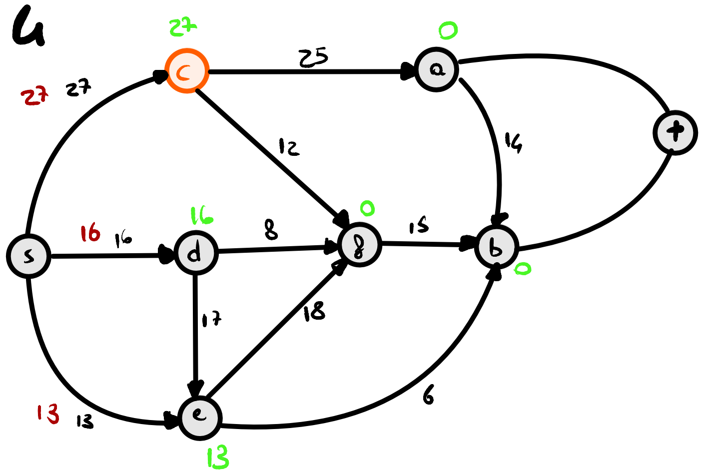
. Trovo G(x) associato e inizializzo Labels d

G(x)



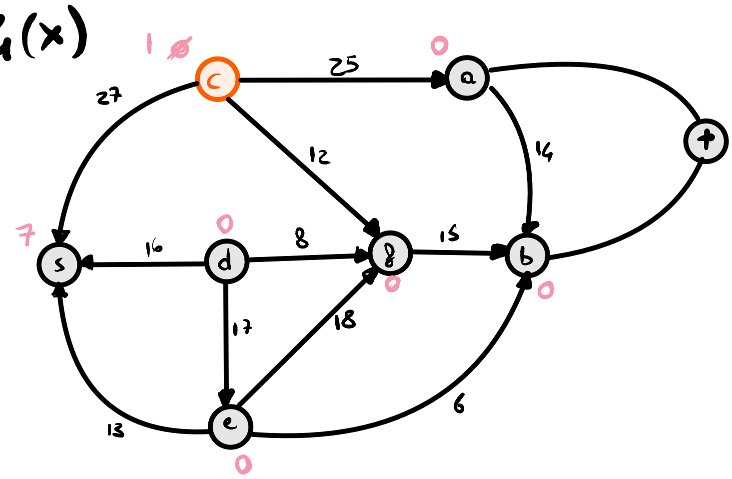
. Seleziono il NODO ATTIVO con massima d

G



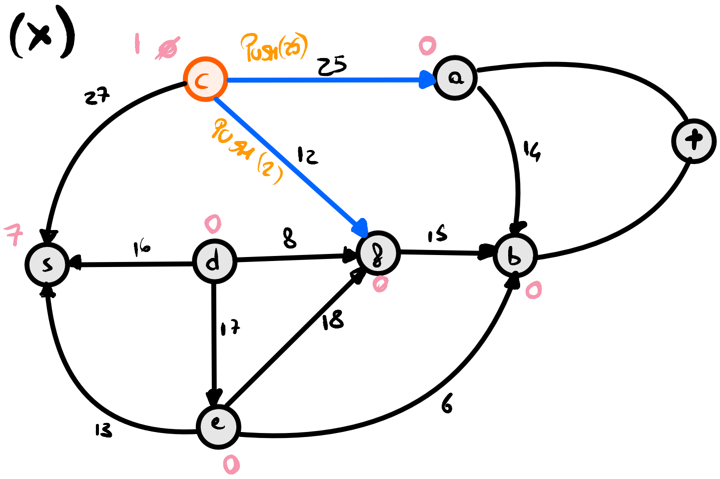
. Non esiste un arco ammissibile dal NODO ATTIVO, quindi effettuo RELABEL

G(x)

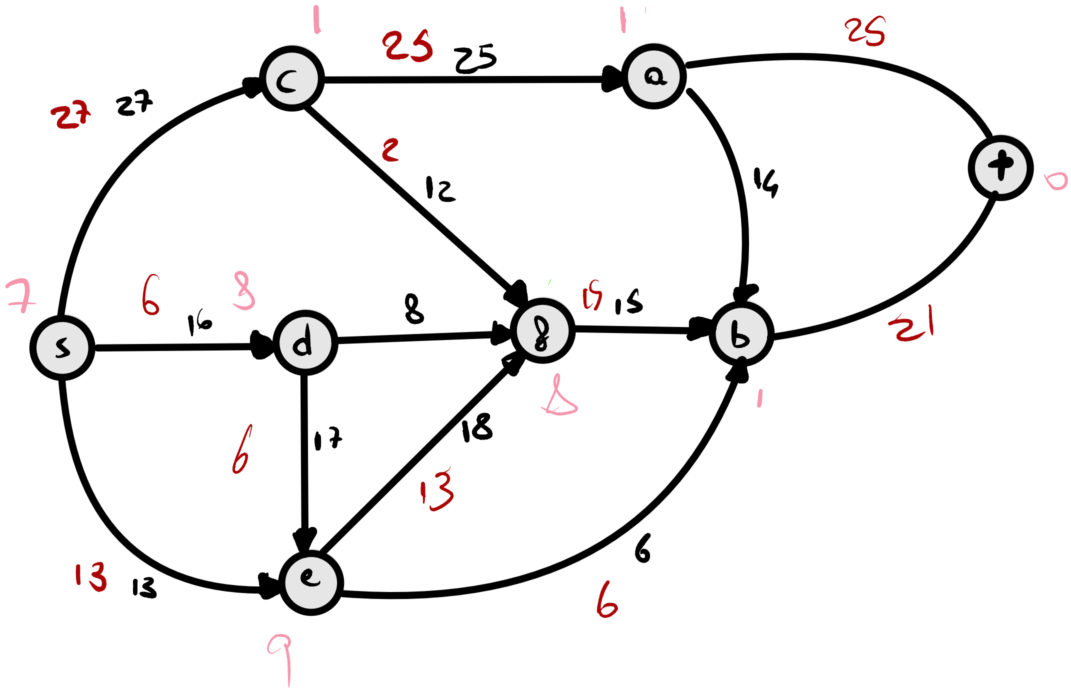


• Esistono ora due archi ammissibili, posso quindi effettuare due PUSH eliminando l'eccesso

G(x)



• Ripeto finché x non è feasible



Network Flows First and Second Midterm
April, 24th 2017

Last Name	________________
First Name	________________
Matricola ID	________________

Question 1

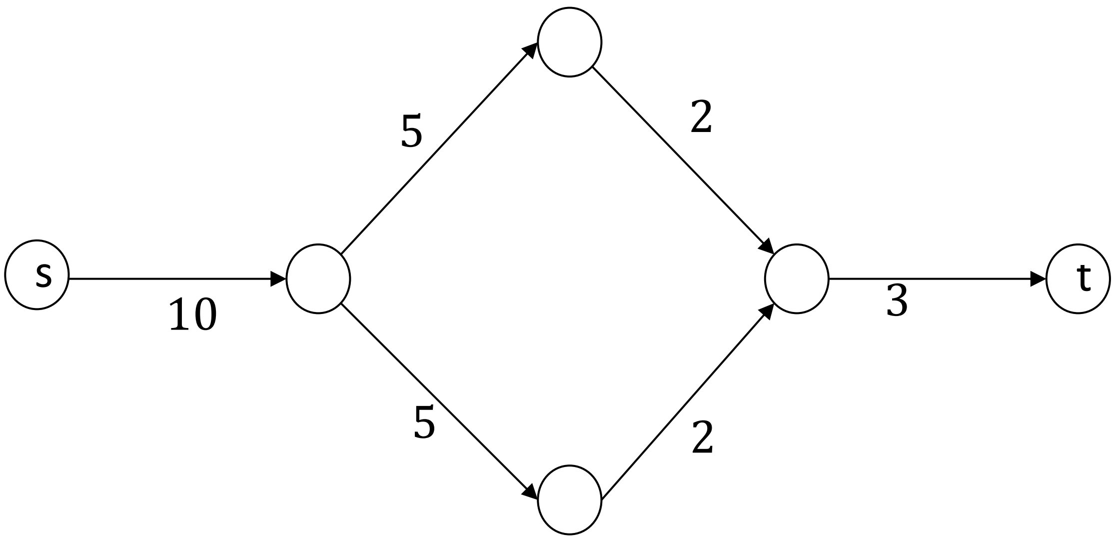
Draw a graph $G=(N,A)$ in which the number of augmentation steps performed by the Ford and Fulkerson algorithm is exactly U, where $U = \max\{u_{ij}\}$ for $(i,j) \in A$
The push-relabel algorithm has a better performance on such a graph?

Question 2

In the push-relabel one can define a rule to choose the next active node being processed. Compare on the following graph the two rules:

1. Choose the active node with the highest distance.
2. Choose the active node with the lowest distance.

(In both rules break ties arbitrarily)



Exercise 1

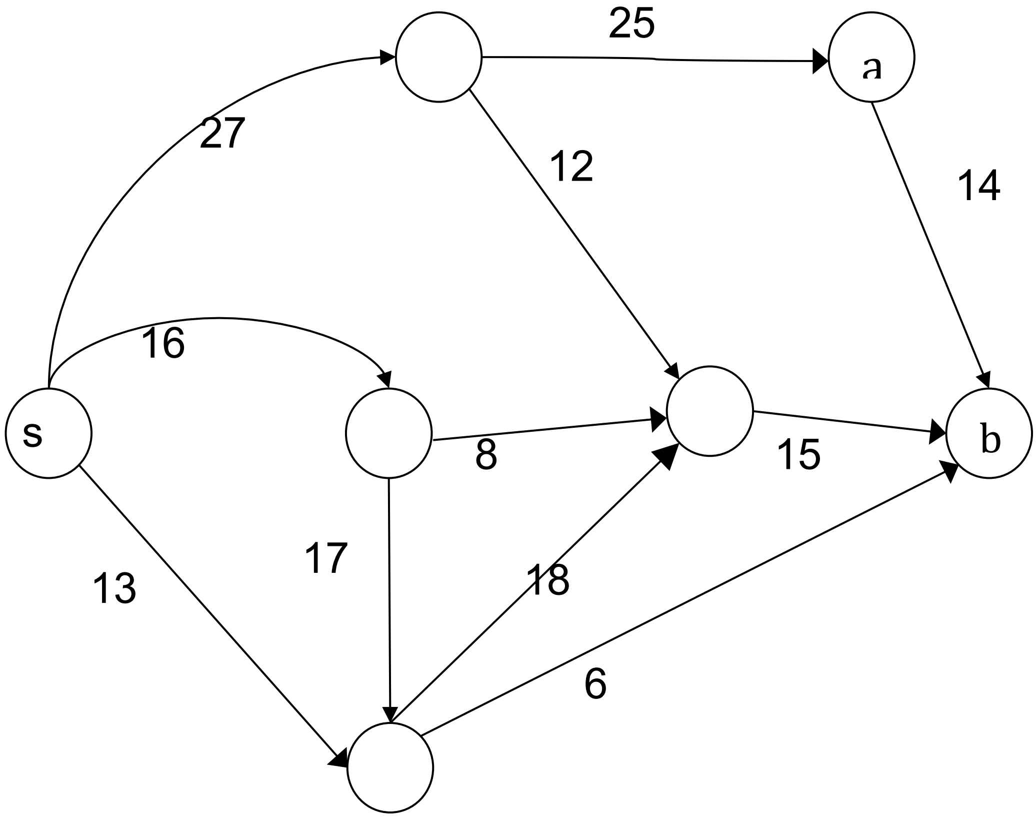
The following graph $G=(N,A)$ represents a logistic distribution network. Node s is a manufacturing plant, origin of the goods and nodes a and b represent warehouses.

1. Evaluate the maximum quantity of goods that can be shipped from the plant to the warehouses by the preflow-push algorithm

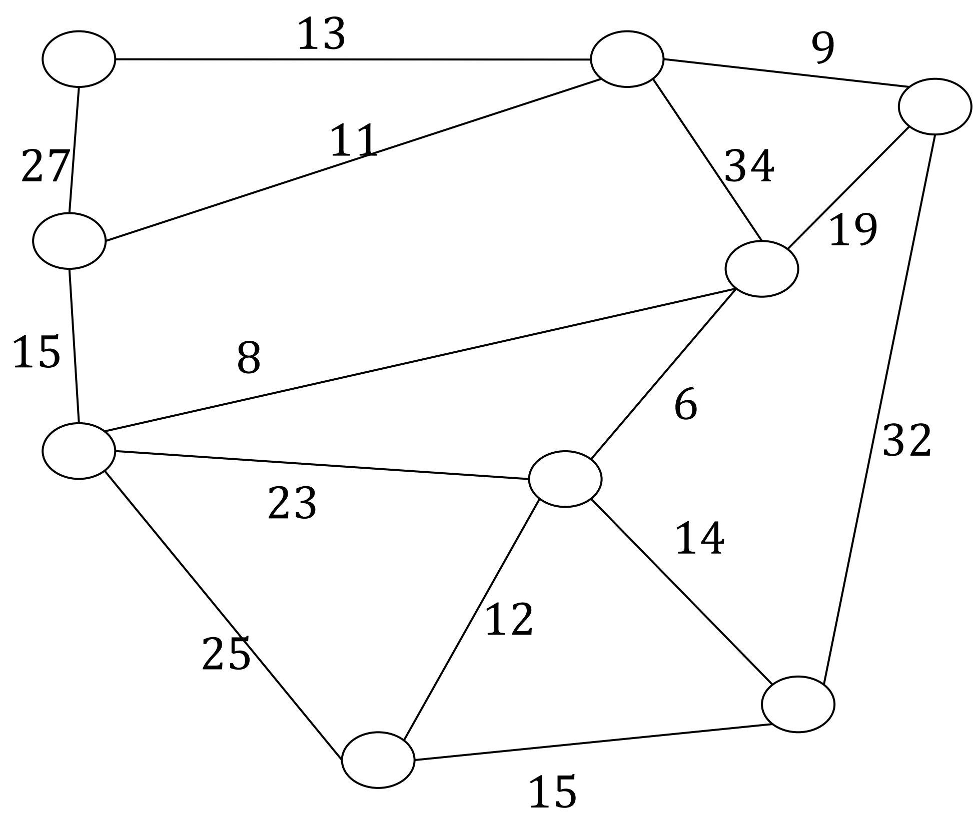
-

Network Flows First and Second Midterm
April, 24th 2017

Last Name ____________________
First Name ____________________
Matricola ID ____________________



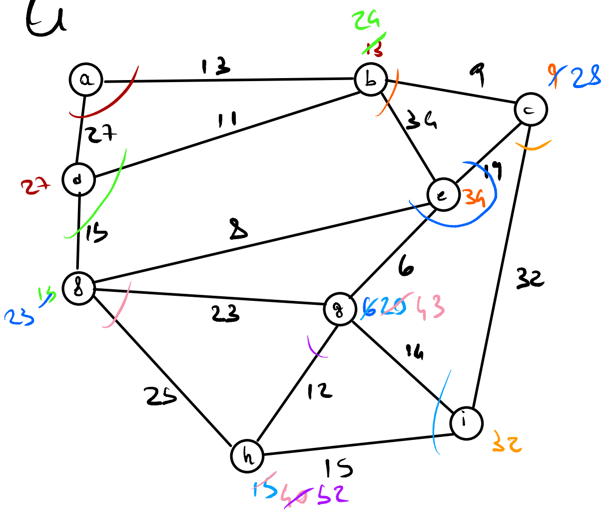
Exercise 3
Find the minimum cut on the following graph:



-

MIN CUT

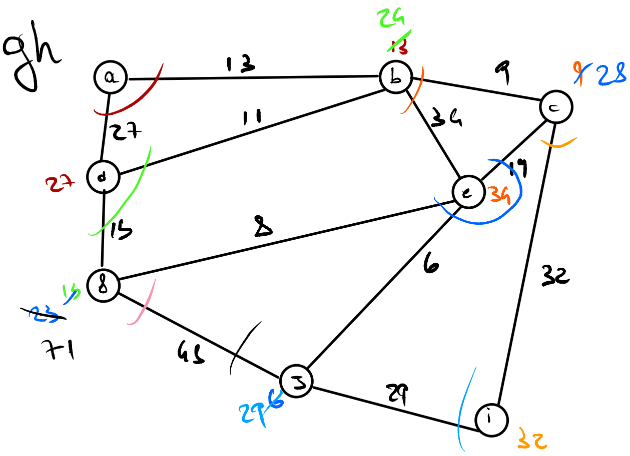
G



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

$\lambda(G, g, h) = 52$

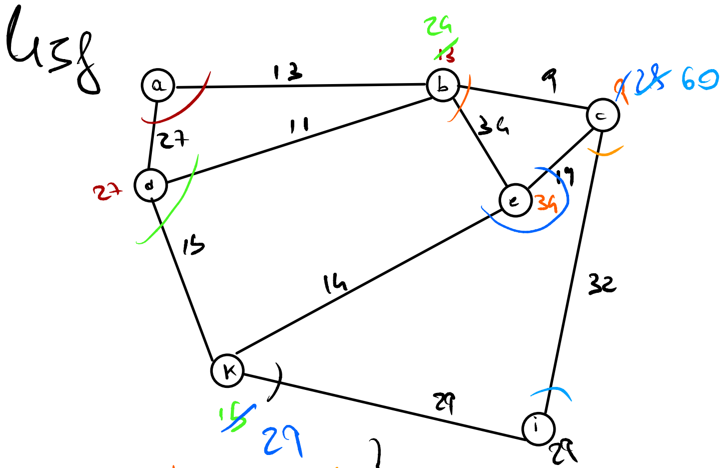
G_{gh}



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

$\lambda(G, s, g) = 71$

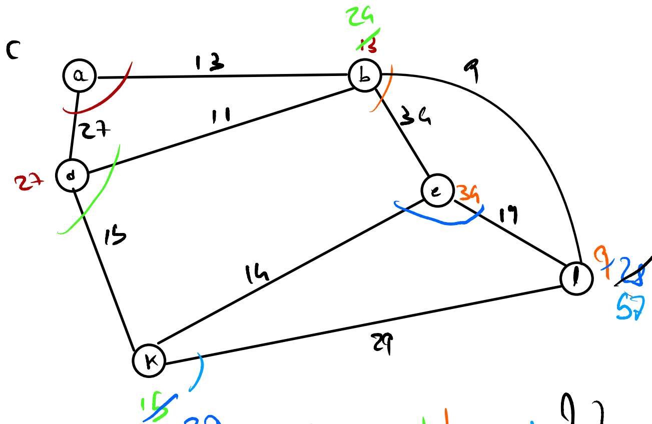
G_{sg}



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

$\lambda(G, i, c) = 60$

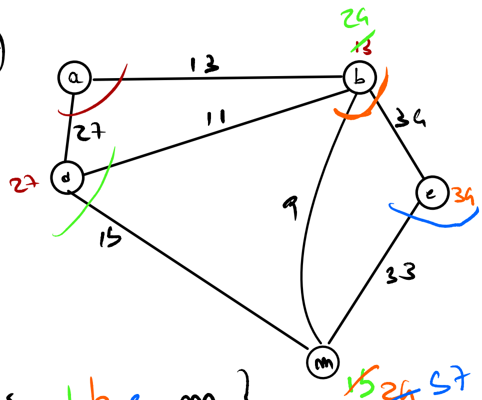
G_{ic}



$O = \{a, d, b, e, k, l\}$

$\lambda(G, k, l) = 57$

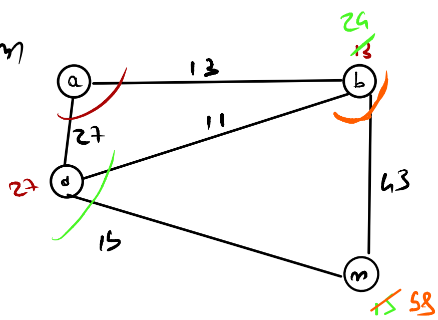
G_{kl}



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

$\lambda(G, e, m) = 57$

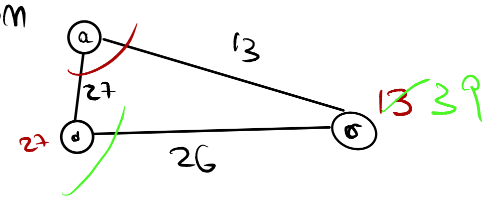
G_{em}



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

$\lambda(G, b, m) = 58$

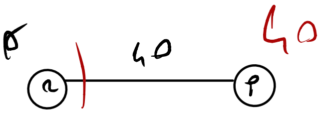
G_{bm}



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

$\lambda(G, d, \varphi) = 39$

$G_{d\varphi}$

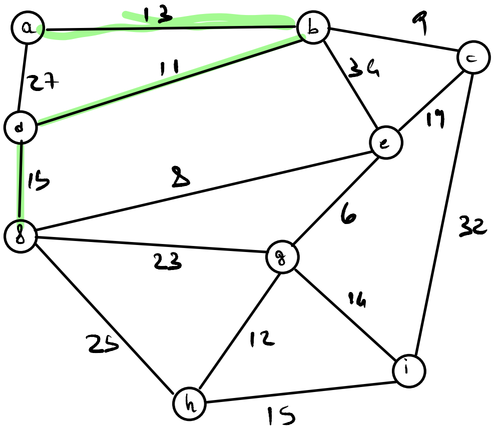


$O = \{a, \varphi\}$

$\lambda(G, a, \varphi) = 40$

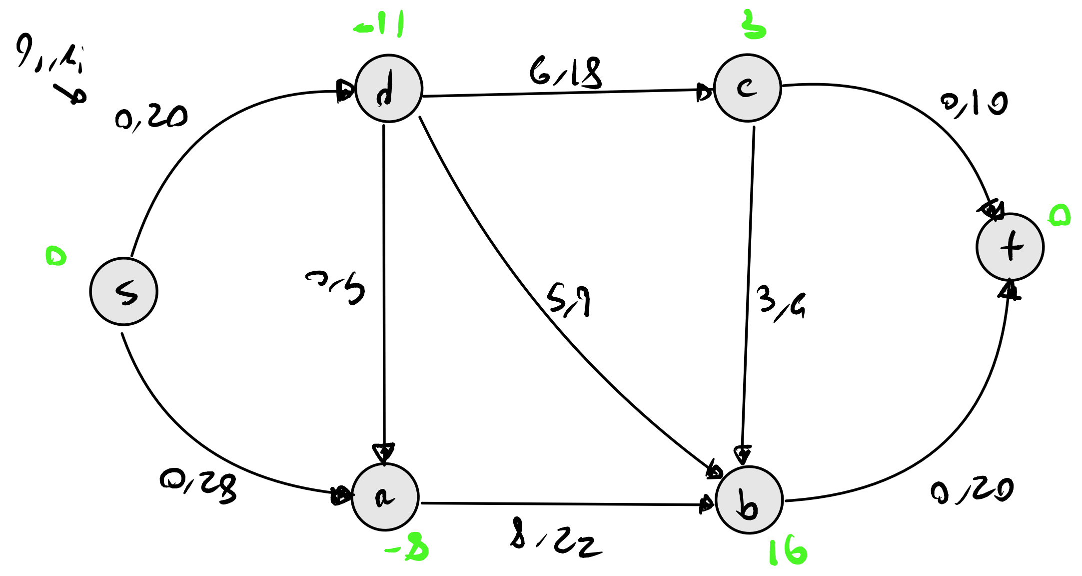
- $\lambda(G) = 39 = \lambda(G, a, d)$

$G \quad \omega t$



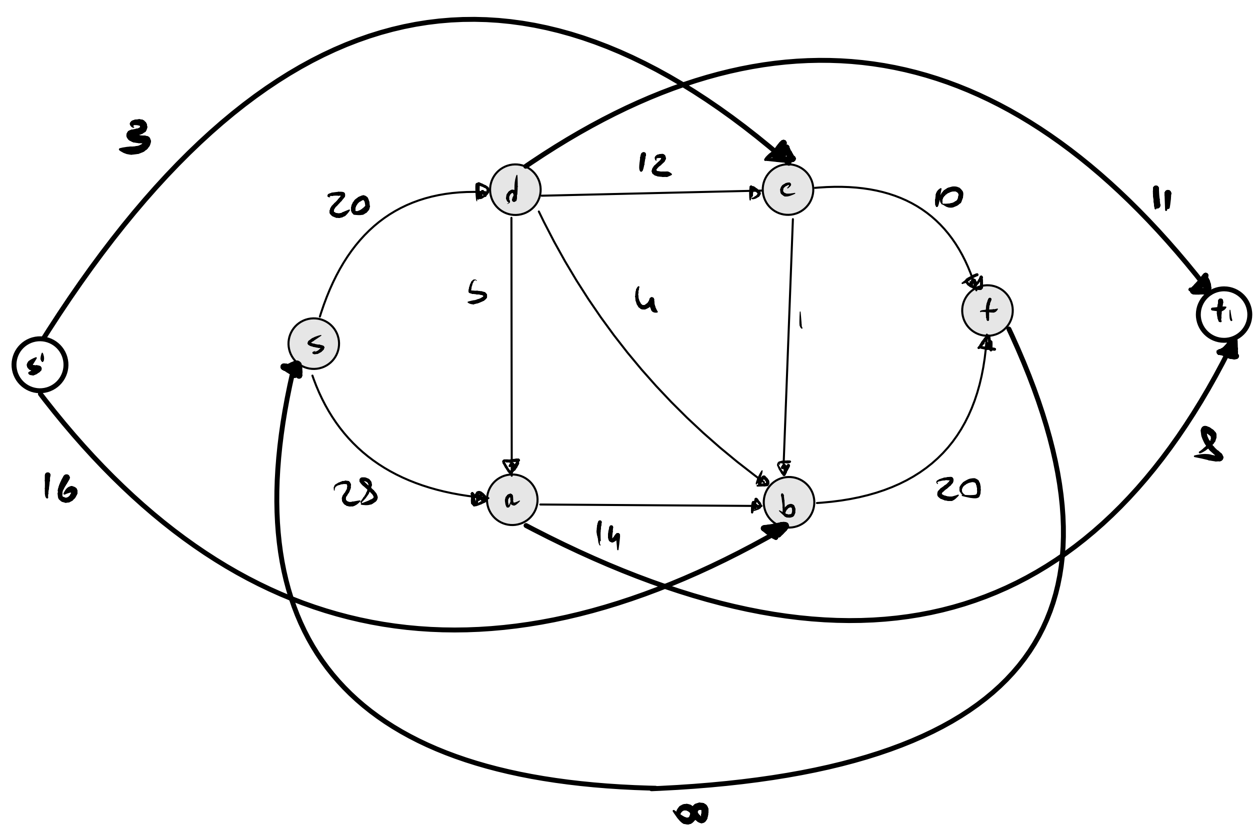
Flow con lower bounds

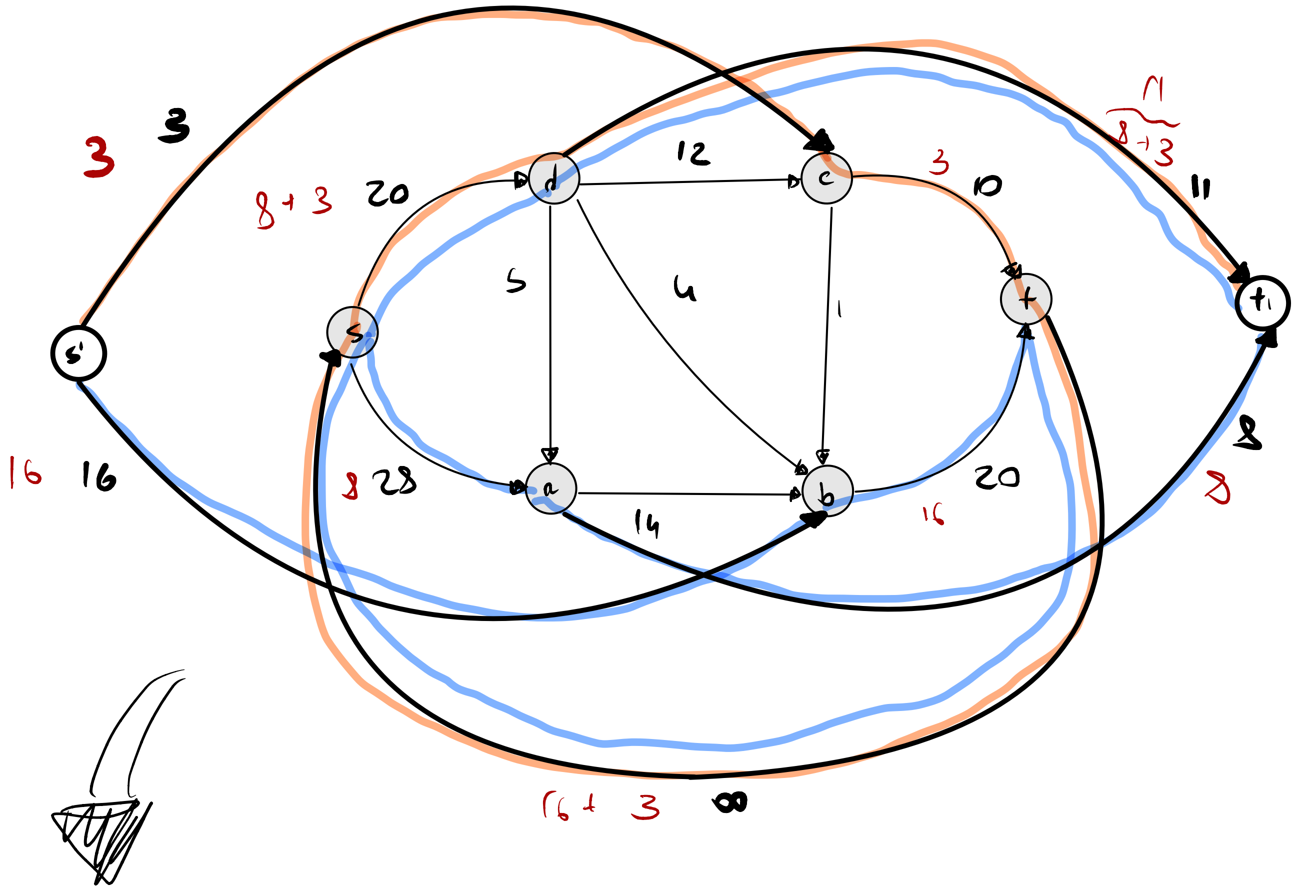
Dato:



(1) Valuta il max (s-t)-flow

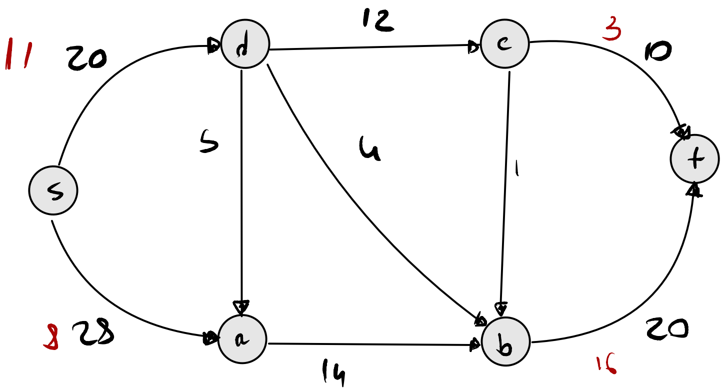
(2) Possiamo incrementare il max (s-t)-flow by adding one arc tra una coppia di nodi che non sia (s,t)





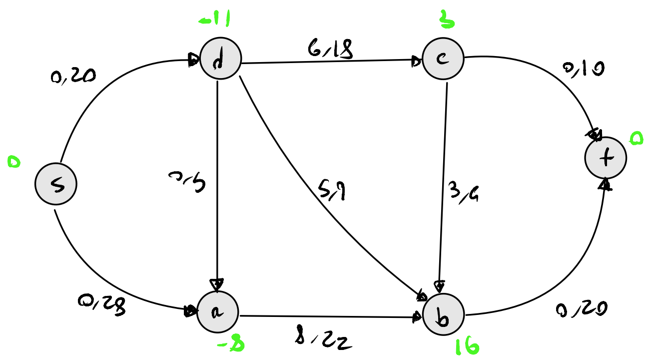
s'
s
d
c
a
b
t
t₁
3 3
8+3 20
12
3 10
11
8+3 11
5
4
1
16 16
8 28
14
20
16
8 8
16+3 ∞

Gtul



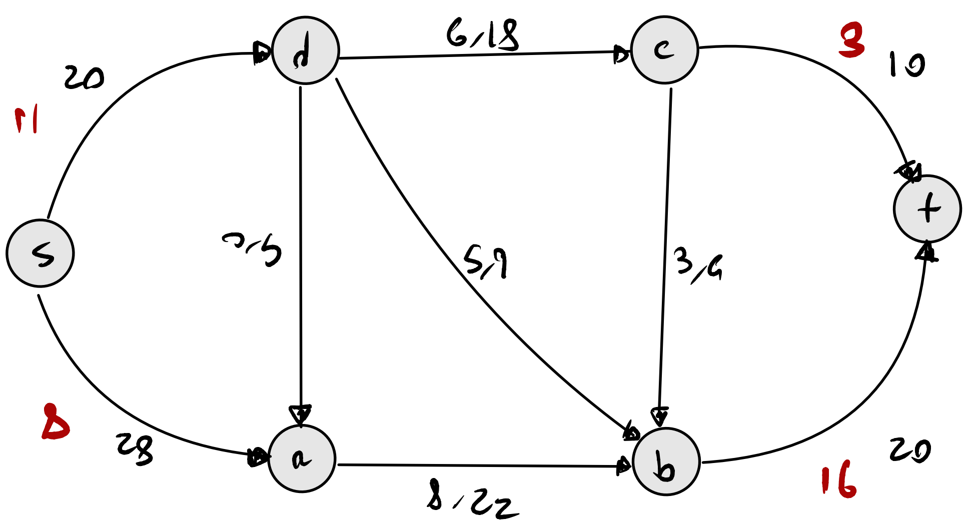
s
d
c
a
b
t
11 20
12
3 10
5
4
1
8 28
14
20
16

G



s
d
c
a
b
t
0
-11
3
0
-8
16
0,20
6/18
0,10
0/5
5/9
3/4
0,28
8/22
0,20

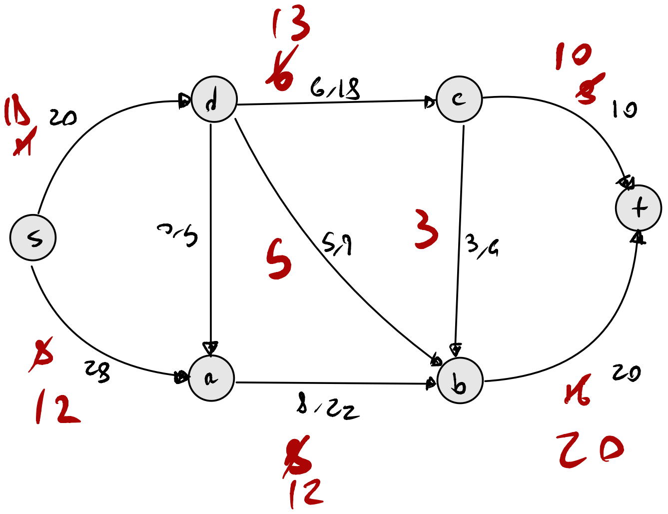
=



s
d
c
a
b
t
11 20
6/18
3 10
0/5
5/9
3/4
8 28
8/22
16 20

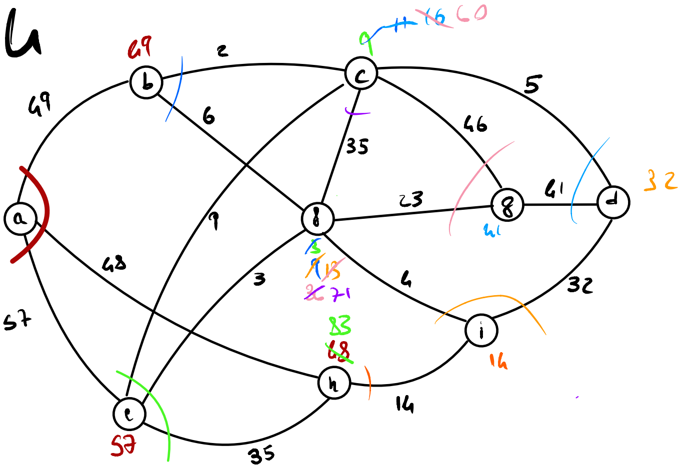
FF

G



MIN CUT

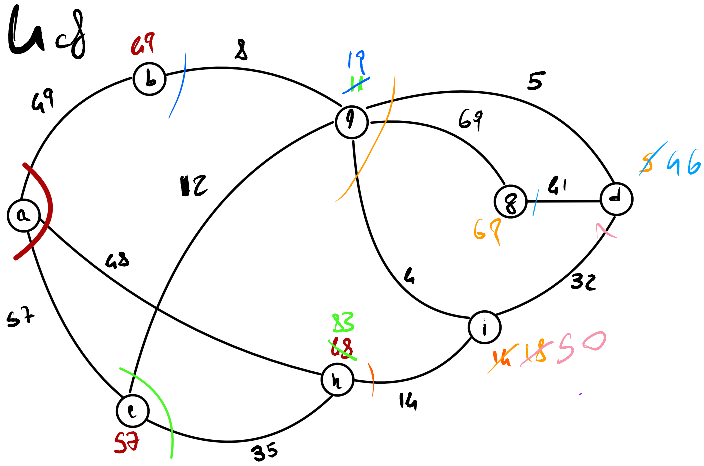
G



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

$\lambda(G, c, g) = 71$

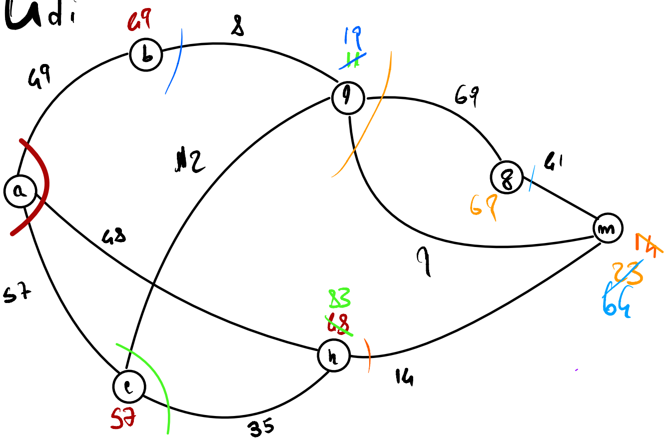
G_{cg}



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

$\lambda(G, d, i) = 50$

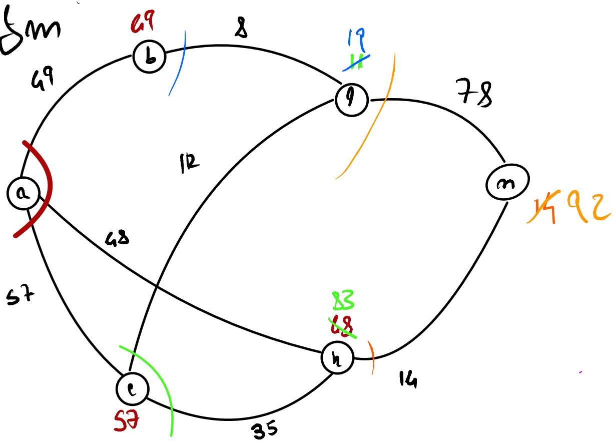
G_{di}



$O = \{a\ e\ h\ b\ f\ g\ m\}$

$\lambda(G, g, m) = 64$

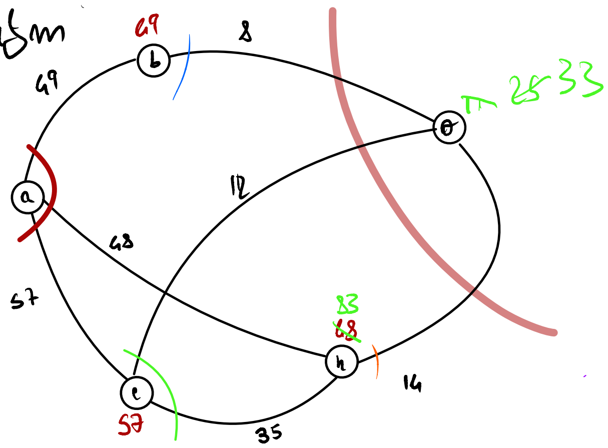
G_{fm}



$O = \{a\ e\ h\ b\ f\ m\}$

$\lambda(G, f, m) = 92$

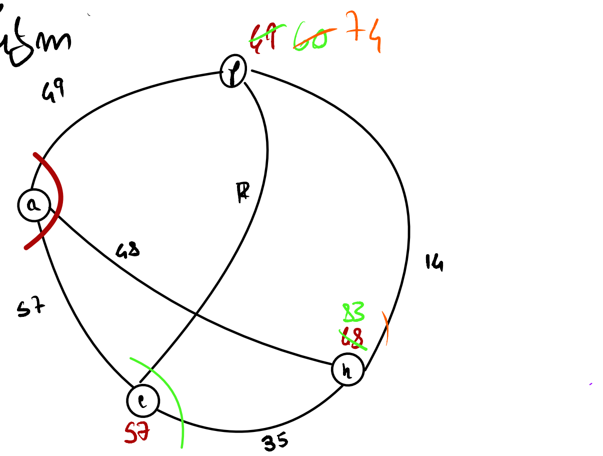
G_{fm}



$O = \{a\ e\ h\ b\ \sigma\}$

$\lambda(G, b, \sigma) = 33$

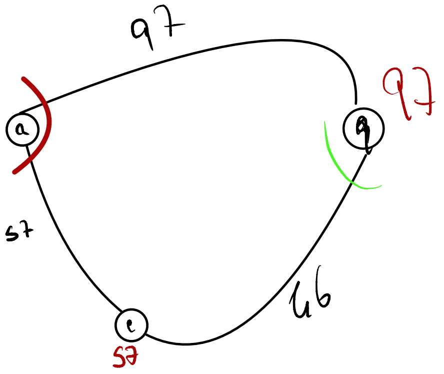
G_{fm}



$O = \{a\ e\ h\ \rho\}$

$\lambda(G, h, \rho) = 74$

G_{sm}



$O = \{a\ q\ c\}$

$\lambda(G, q, c') = 103$

G_{sm}

$O = \{a\ q\}$

$\lambda(G, q, c') = 154$

G