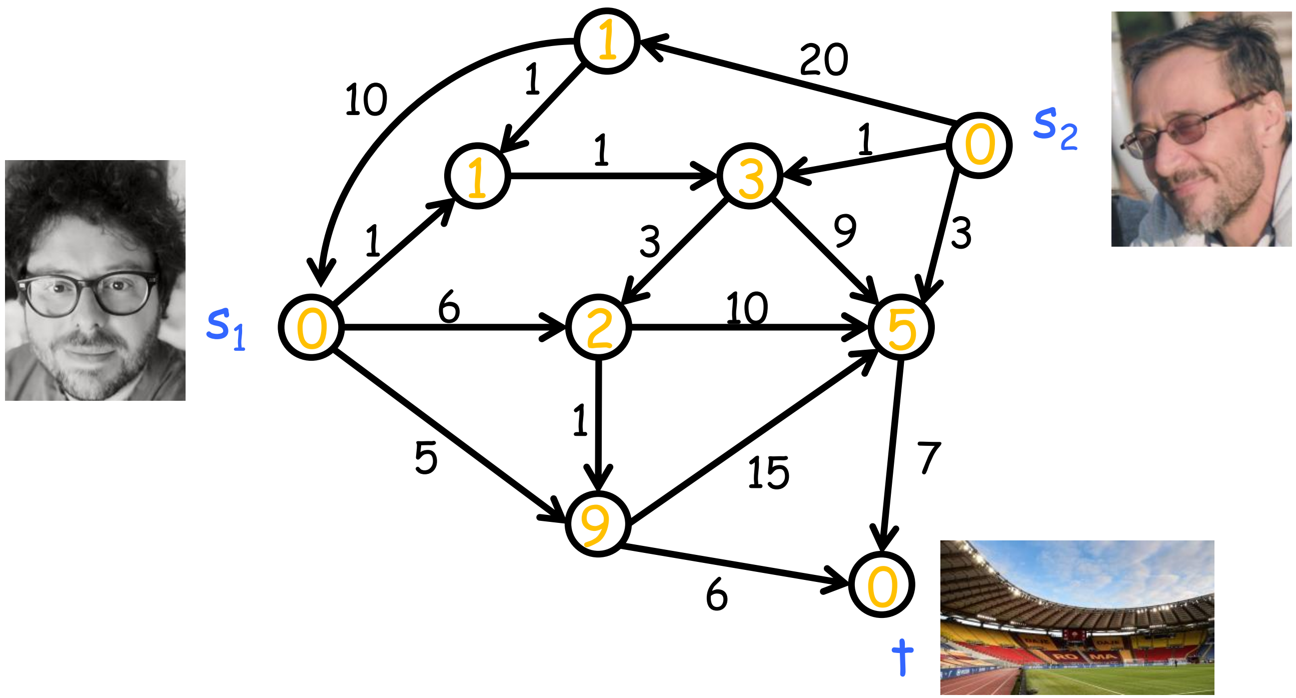
Esercitazione
18 dicembre 2024

Esercizio 1:

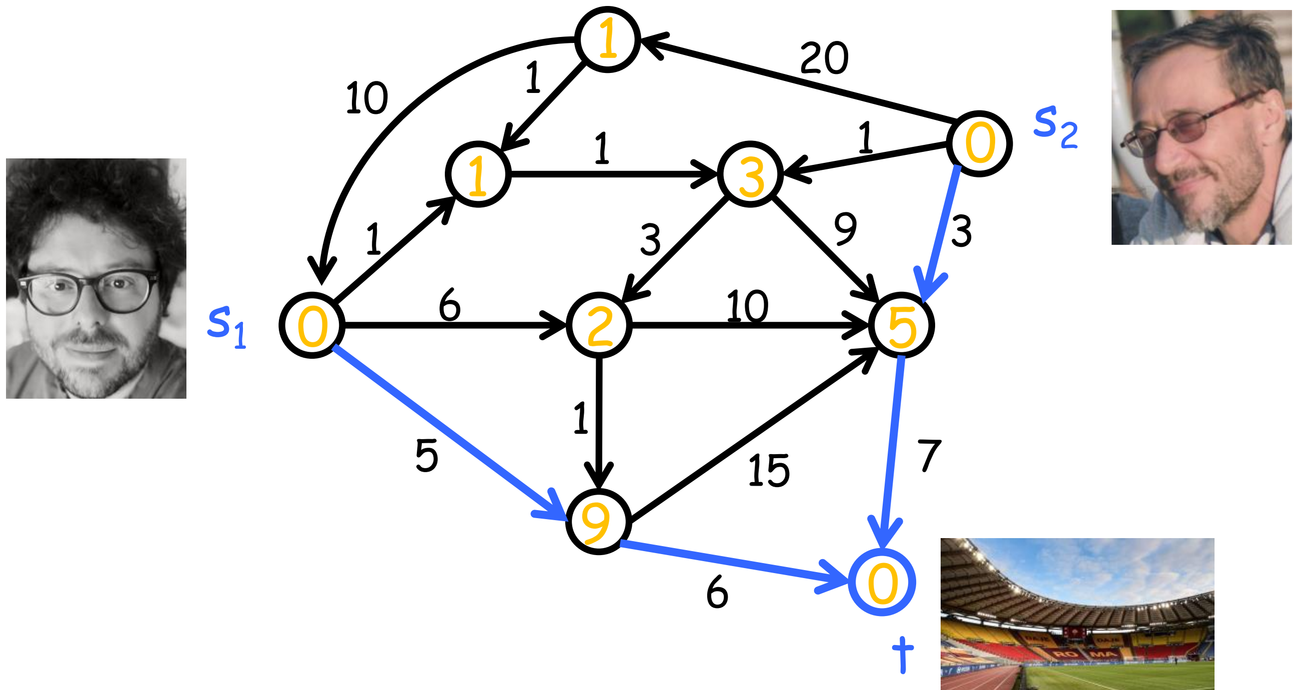
Gualà e Clementi vanno a vedere la (magggica) Roma

Problema (Gualà e Clementi vanno a vedere la (maggggica) Roma)

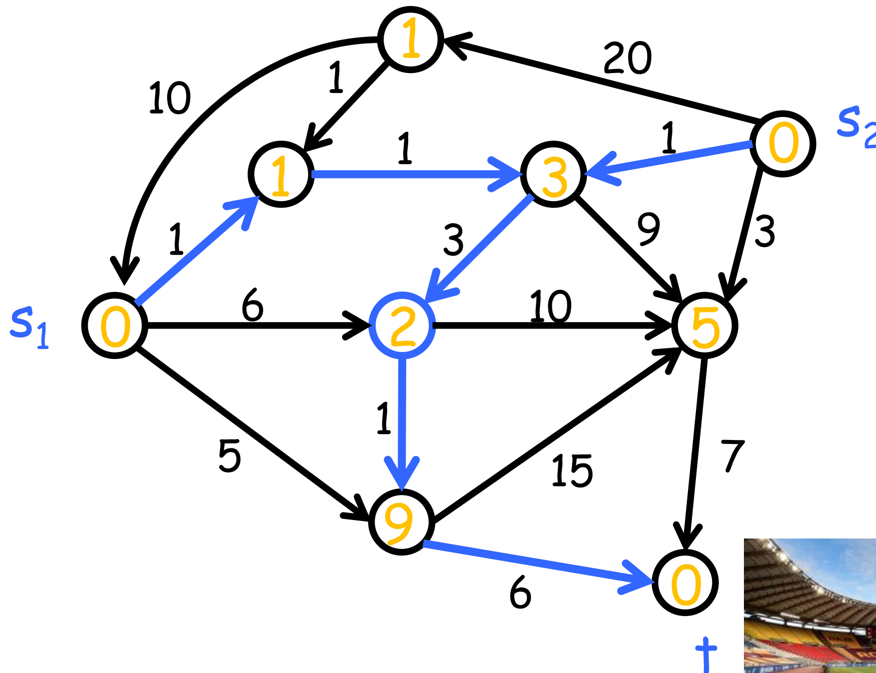
Una città è modellata come un grafo diretto e pesato $G = (V, E, w)$, dove ad ogni arco $e \in E$ è associato un peso $w(e) \geq 0$ che rappresenta il costo, in termini di benzina consumata, per attraversare l'arco (strada) e. In questa città, i vostri docenti del corso di algoritmi, Gualà e Clementi, vogliono andare a vedere la partita della Roma allo stadio, che si trova nel nodo t. Loro sono rispettivamente nei nodi s_1 e s_2, e possiedono una macchina ciascuno. Volendo, possono incontrarsi in un nodo del grafo, parcheggiare una delle due macchine, e proseguire insieme. Ma di solito in questa città, parcheggiare costa. Per ogni nodo v, dunque, conoscono il costo $c(v)$ del parcheggio presente in v (si può assumere per semplicità che $c(s_1) = c(s_2) = c(t) = 0$). Progettate un algoritmo che in tempo $O(m + n \log n)$ calcoli la soluzione che Gualà e Clementi devono adottare per spendere complessivamente il meno possibile in termini di corso della benzina più costo del parcheggio.



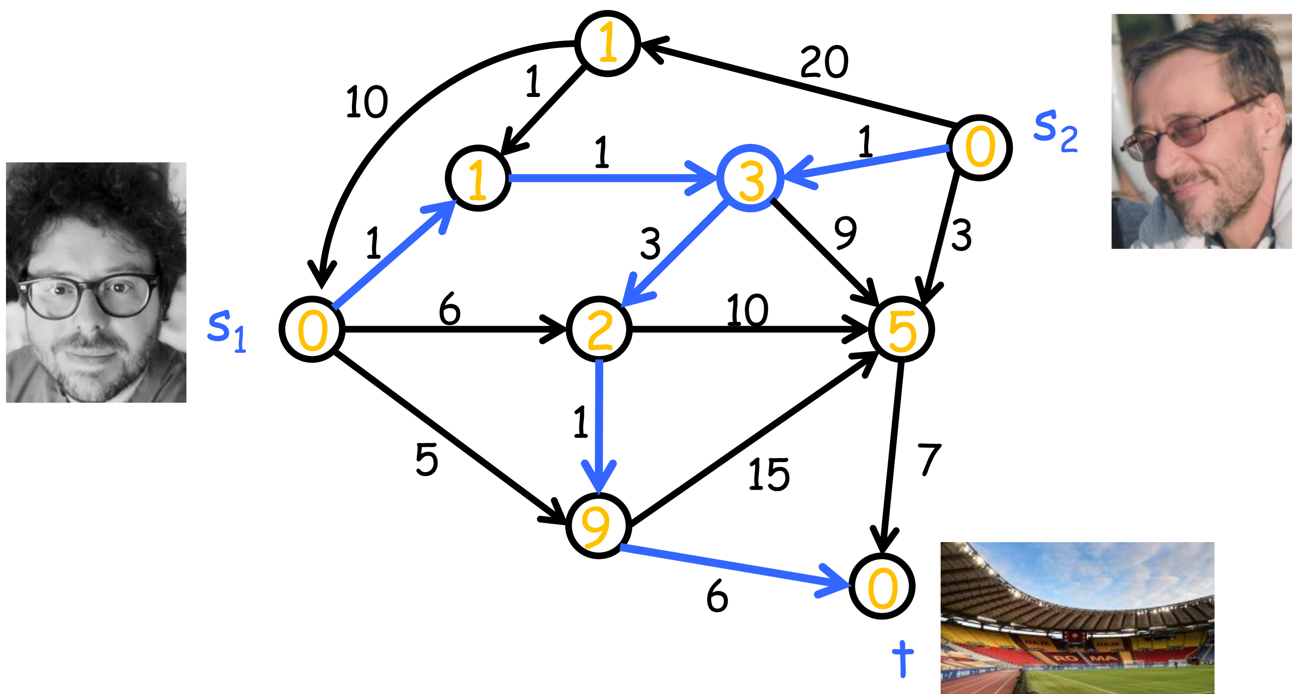
s_1
s_2
t
0
1
1
3
0
2
5
9
0
20
10
1
1
1
1
6
3
9
3
10
5
1
15
7
6



costo=11+10+0=21



costo=5+4+2+7=18

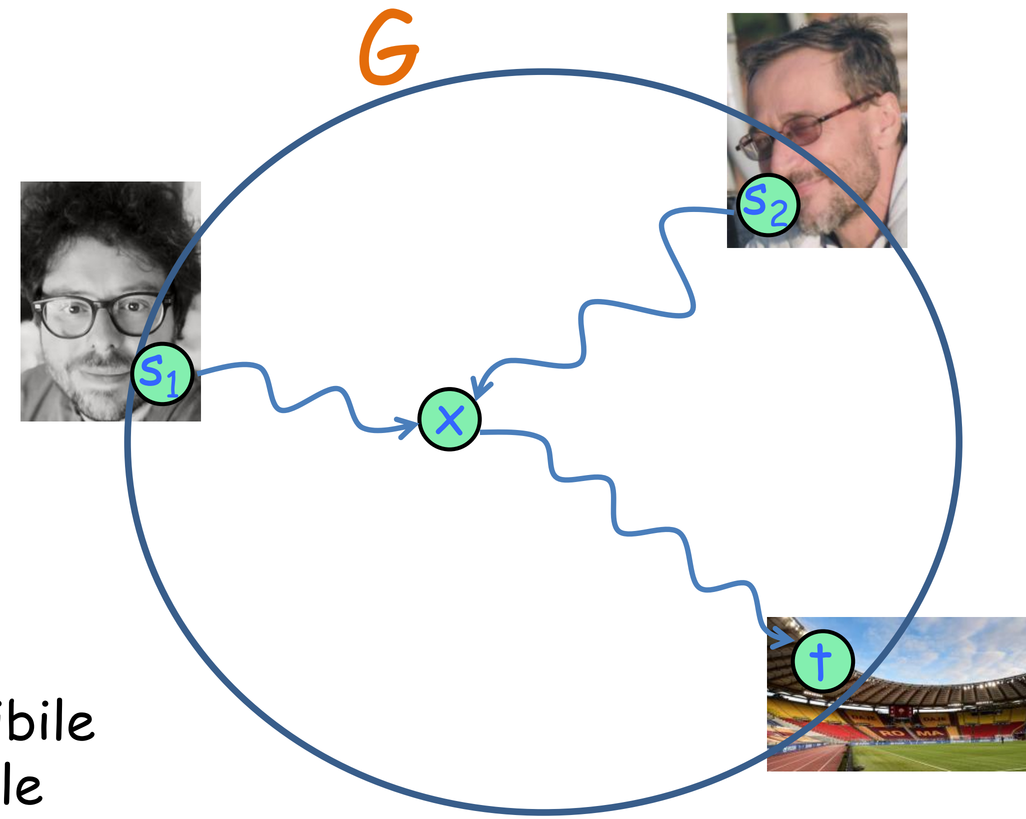


costo=2+1+3+10=16

idea: "indovinare" il nodo x dove Gualà e Clementi si incontrano per lasciare una macchina

$$\mathrm{cost}(x) := d(s_1,x)+d(s_2,x)+c(x)+d(x,t)$$

$\mathrm{cost}(x)$: costo totale se Gualà e Clementi parcheggiano una macchina in x



osservazione: $\mathrm{cost}(x)$ è diponibile in tempo cosante se ho tutte le distanze a singola sorgente *da* s_1,s_2, e *verso* t

corretto?

sì: provo tutti gli x

$cost(x) := d(s_1,x)+d(s_2,x)+c(x)+d(x,t)$

complessità?

Algoritmo

- calcola distanze/SPT con sorgenti s_1,s_2 } $O(m+n\log n)$
- calcola distanze/SPT verso t (calcolando distanze/SPT con sorgete t nel grafo con archi girati al contrario) } $O(m+n\log n)$
- $z=\arg\min_{x\in V} cost(x)$ } $O(n)$
- restituisci cost(z) } $O(1)$

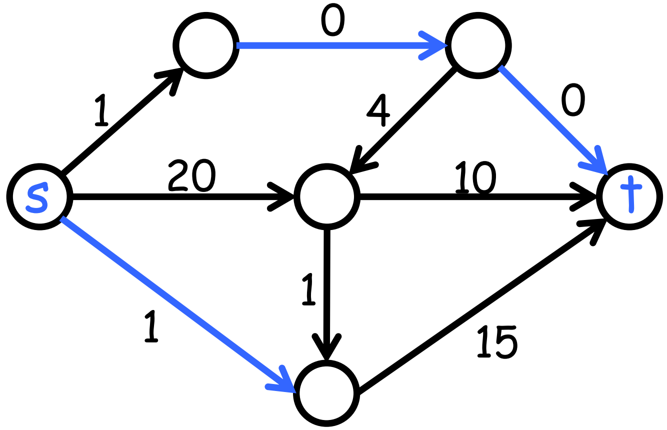
$\Rightarrow$ $O(m+n \log n)$

Esercizio 2:

(Ex 3, PS 2020).

Input: -grafo orientato G=(V,E,w) con pesi non negativi
-B⊆E sottoinsieme di archi blu
-k intero, s,t ∈V

Output: un cammino di costo minimo da s a t che usa al più k archi blu



k=1

idea 1: "indovinare" i k archi blu della soluzione.

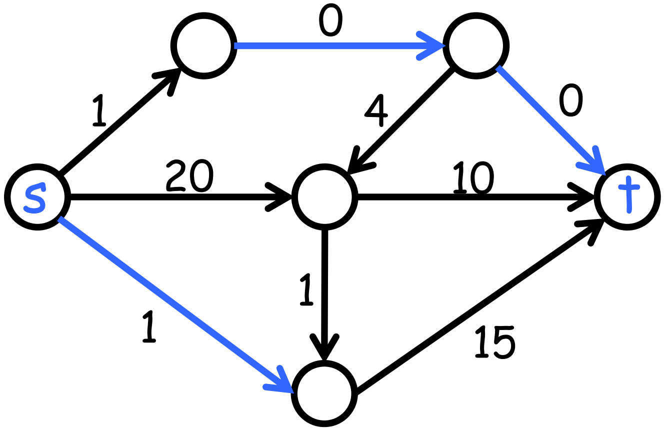
$\overline{G} = (V, E \setminus B)$

per ogni k-tupla F di archi in B:

- calcola il cammino minimo da s a t nel grafo $\overline{G}$+F

restituisci il miglior cammino trovato

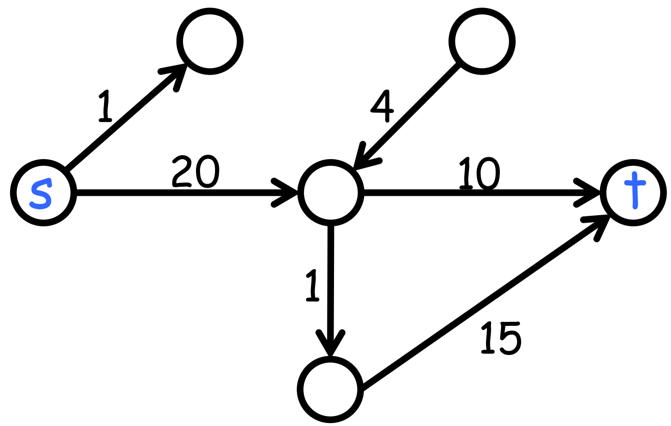
k=1



idea 1: “indovinare” i k archi blu della soluzione.

$\overline{G}$ = (V,E\B)
per ogni k-tupla F di archi in B:
  - calcola il cammino minimo da s a t nel grafo $\overline{G}$+F
restituisci il miglior cammino trovato

k=1



idea 1: “indovinare” i k archi blu della soluzione.

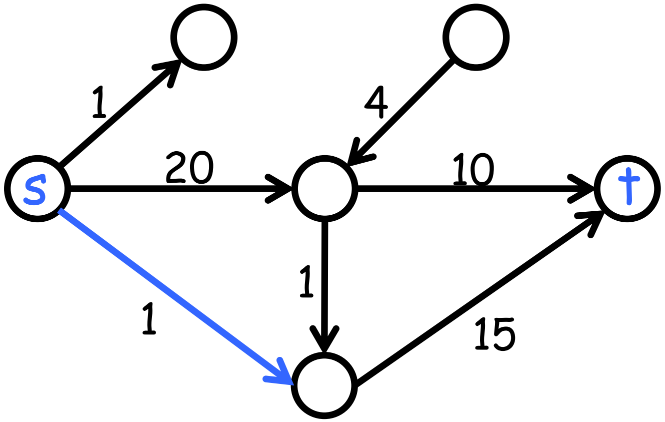
$\overline{G}$ = (V,E\B)

per ogni k-tupla F di archi in B:

- calcola il cammino minimo da s a t nel grafo $\overline{G}$+F

restituisci il miglior cammino trovato

k=1



idea 1: “indovinare” i k archi blu della soluzione.

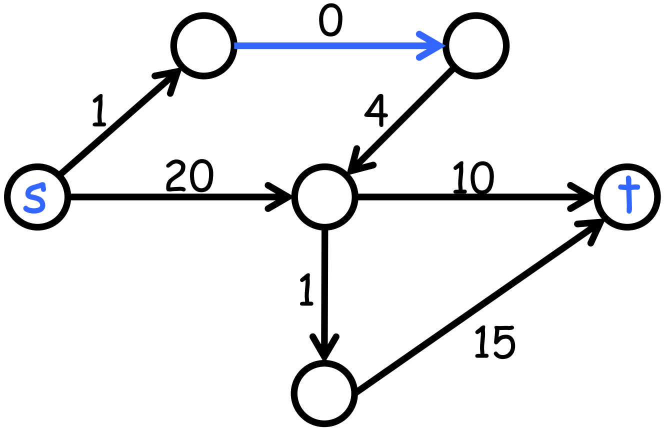
$\overline{G}$ = (V,E\B)

per ogni k-tupla F di archi in B:

- calcola il cammino minimo da s a t nel grafo $\overline{G}$+F

restituisci il miglior cammino trovato

k=1

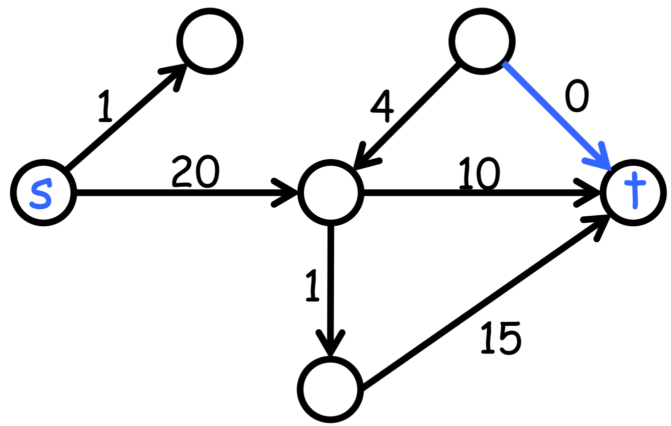


idea 1: "indovinare" i k archi blu della soluzione.

$\overline{G}$ = (V,E\B)
per ogni k-tupla F di archi in B:
- calcola il cammino minimo da s a t nel grafo $\overline{G}$+F

restituisci il miglior cammino trovato

k=1



idea 1: "indovinare" i k archi blu della soluzione.

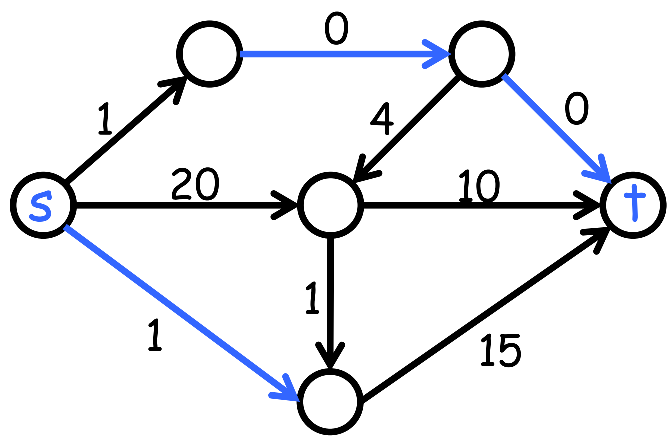
$\overline{G}$ = (V,E\B)
per ogni k-tupla F di archi in B:
- calcola il cammino minimo da s a t nel grafo $\overline{G}$+F

restituisci il miglior cammino trovato

correttezza?

k=1

-ogni cammino calcolato è un cammino ammissibile;
-quando guardo la k-tupla usata dalla soluzione (o un sovrainsieme) il cammino calcolato è quello ottimo cercato

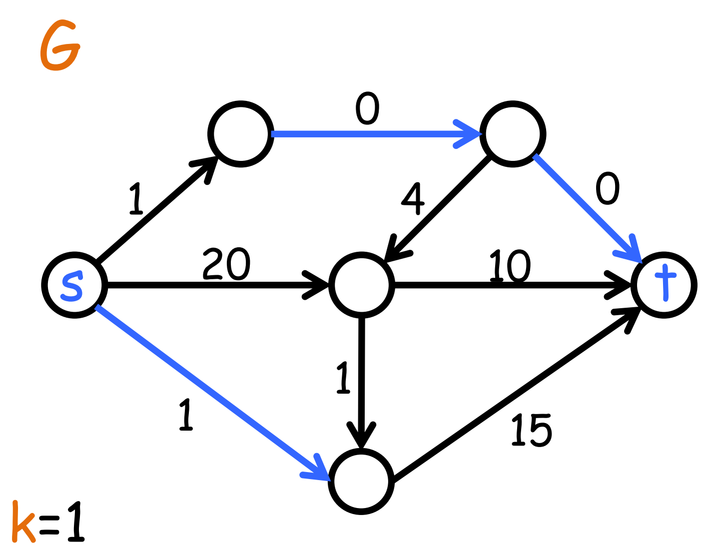


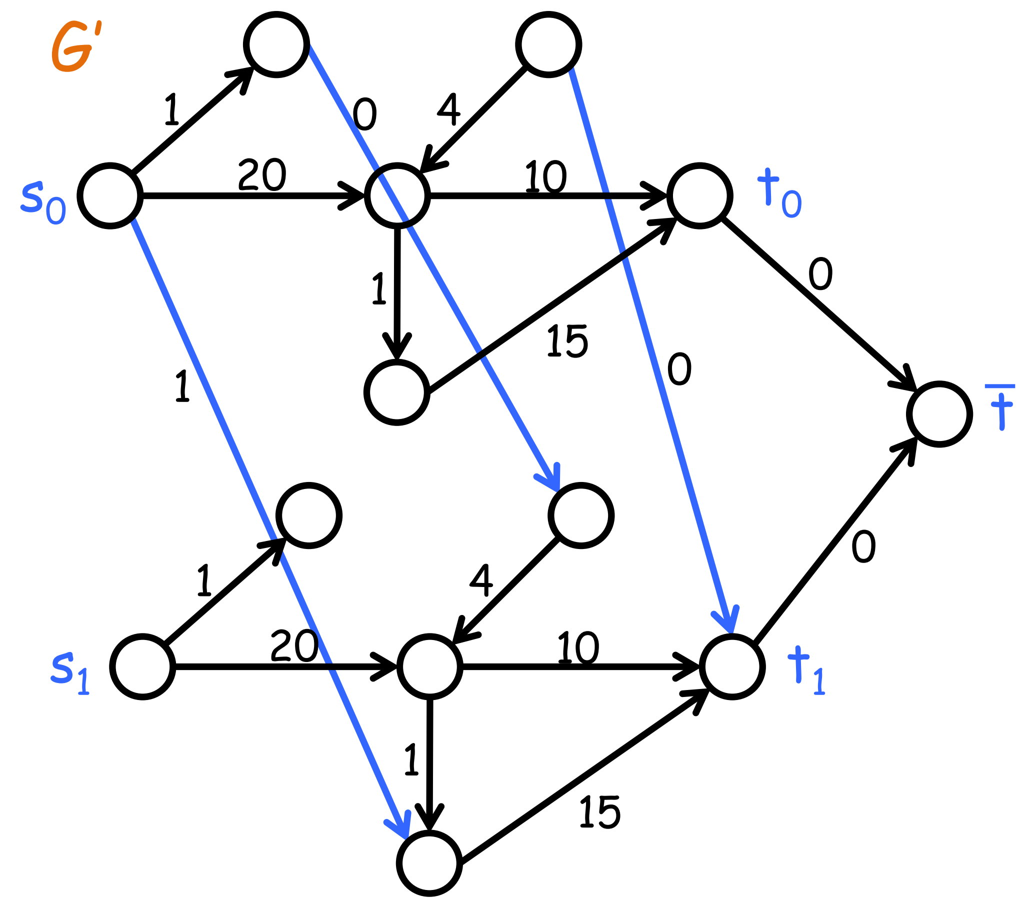
complessità?

$O(|B|^k(m+n \log n))$

idea 2: ridurre il problema al calcolo di un cammino minimo su un opportuno grafo ausiliario G'

- G' fatto "a livelli"
- ogni volta che uso un arco blu sono costretto a cambiare livello
- #livelli dipende da k





cerco il cammino minimo da s_0 a $\bar{t}$ in G'

nodi:

definizione di G'

-per ogni $v \in V$ ho k+1 nodi $v_0, v_1, \dots v_k$

-un nodo $\bar{t}$

archi:

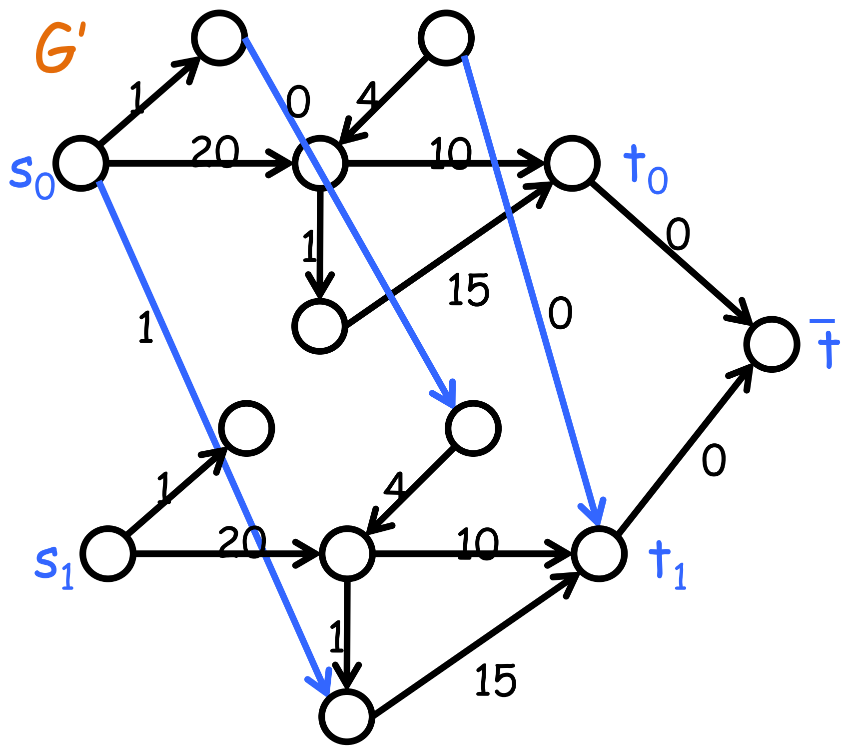
-per ogni arco (u,v) non blu in G ho gli archi (u_i,v_i), i=0,1,…k di peso w(u,v)

-per ogni arco (u,v) blu in G ho gli archi (u_i,v_{i+1}), i=0,1,…k-1 di peso w(u,v)

-ho archi $(t_i,\bar{t})$, i=0,1,…k, di peso 0

soluzione cercata: cammino minimo in G' da s_0 a $\bar{t}$

opportunamente riconvertita in G



correttezza:

Proprietà:

Esiste un cammino in G da s a t che usa al più k archi blu di costo W se e soltanto se esiste un cammino in G' da s_0 a $\overline{t}$ di costo W.

complessità:

dimensione di G':

$n' = n(k+1) + 1 = \Theta(nk)$

$m' \leq (k+1)m + k = \Theta(mk)$

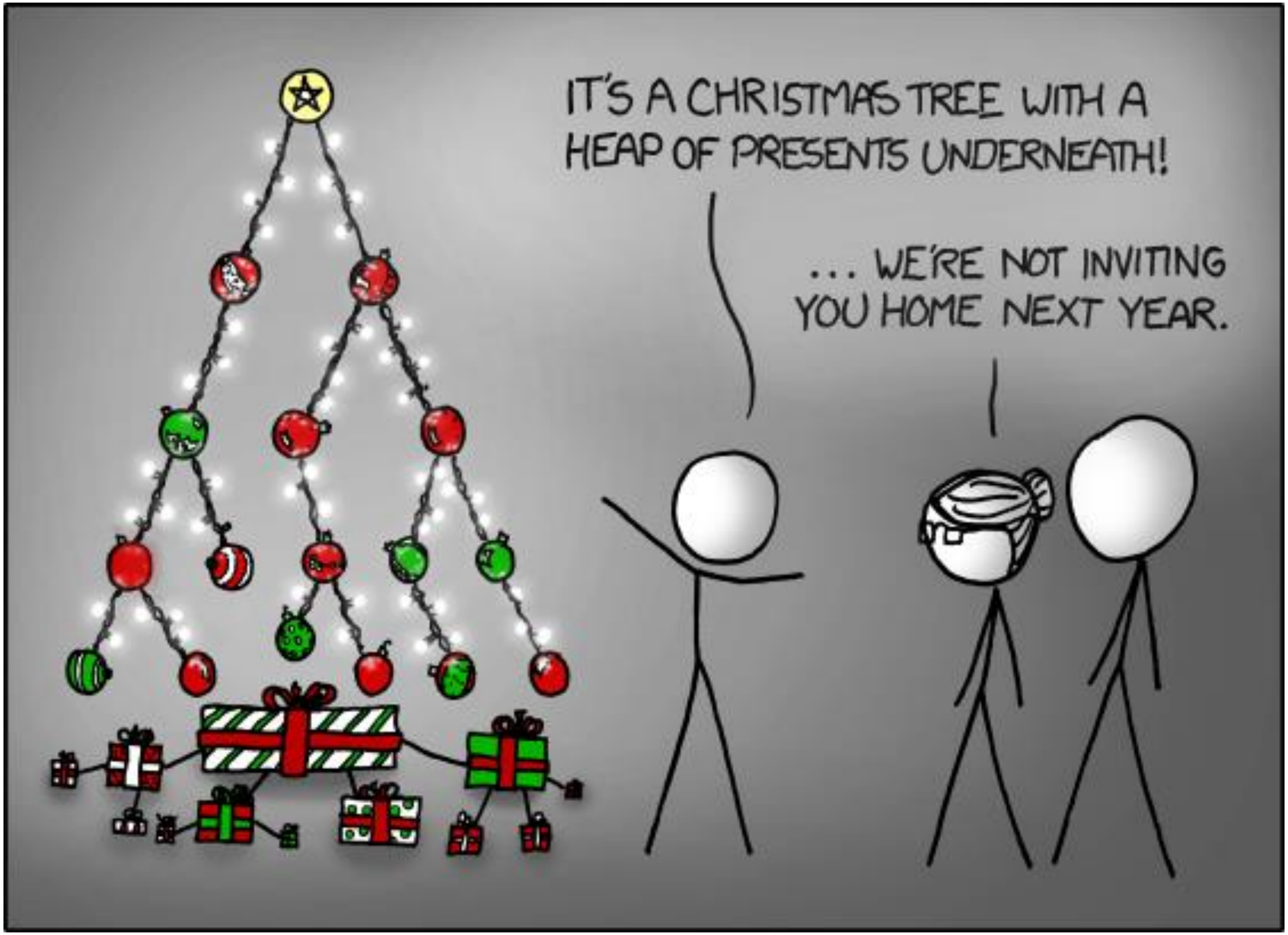
-costruzione di G':

$O(m'+n') = O(k(m+n))$

-calcolo cammino minimo in G':

$O(m'+n' \log n') = O(k(m+n \log n))$

$O(k(m+n \log n))$


IT'S A CHRISTMAS TREE WITH A HEAP OF PRESENTS UNDERNEATH!
... WE'RE NOT INVITING YOU HOME NEXT YEAR.

buone feste!