ESERCIZIO 2

Definizione Dato un grafo non orientato $G = (V, E)$ con funzione peso $w : E \to \mathbb{R}$ e dato un sottoinsieme U di V, il sottografo di (G, w) INDOTTO DA U è il grafo pesato ottenuto rimuovendo da G tutti i nodi non appartenenti a U e tutti gli archi che toccano qualche nodo non appartenente a U.

Sia quindi $G = (V, E)$ un grafo connesso non orientato con funzione peso $w : E \to \mathbb{R}$.

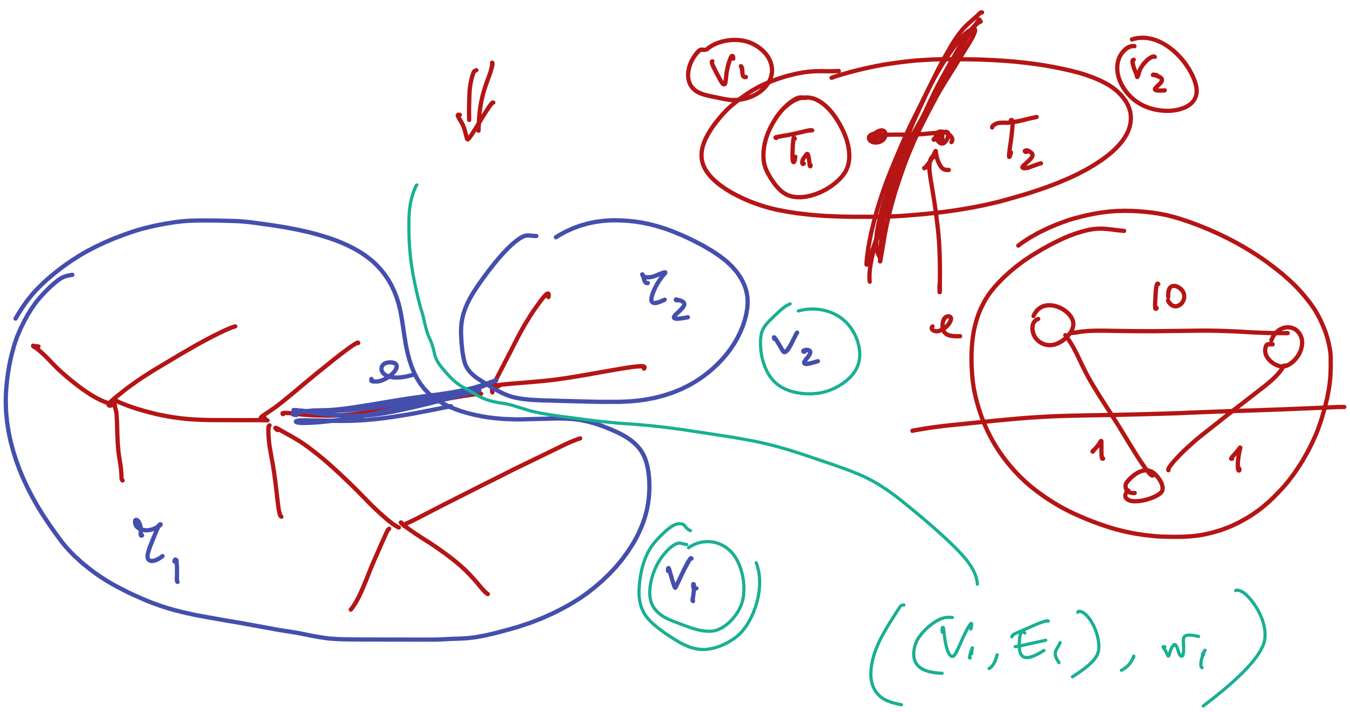
Per ciascuna delle seguenti asserzioni, stabilire se è necessariamente vera oppure no, motivando adeguatamente le risposte.

(A) Sia $\mathcal{T} = (V, T)$ un albero ricoprente minimo di (G, w) e sia $e \in T$ un suo arco. Rimuovendo e da $\mathcal{T}$ si formano due alberi $\mathcal{T}_1$ e $\mathcal{T}_2$ insistenti rispettivamente sugli insiemi di nodi V_1 e V_2. Allora,

- e è un arco di peso minimo che attraversa il taglio (V_1, V_2) di G;
- $\mathcal{T}_i$ è un albero ricoprente minimo del sottografo di (G, w) indotto da V_i, per $i = 1, 2$.

tale che i sottografi di G indotti da V1 e da V2 siano connessi

(B) Sia (V_1, V_2) un taglio di G, sia e un arco di peso minimo che attraversa il taglio (V_1, V_2), e sia $\mathcal{T}_i = (V_i, T_i)$ un albero ricoprente minimo del sottografo di (G, w) indotto da V_i, per $i = 1, 2$. Allora, il grafo $(V, T_1 \cup T_2 \cup \{e\})$ è un albero ricoprente minimo di (G, w).



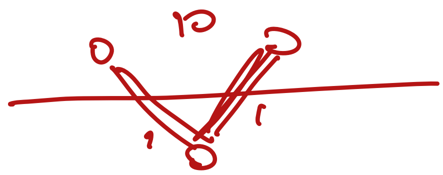
ESERCIZIO 1

Dato un grafo $G = (V, E)$ non orientato e connesso, con funzione peso $w : E \to \mathbb{R}$, si consideri la seguente procedura ricorsiva:

> **procedure** Quick_Something(V, E)
>   **if** $|V| \leq 1$ **then**
>   **return** $\emptyset$
>   **else**
>   - sia (V_1, V_2) un taglio di (V, E) tale che i sottografi (V_1, E_1) e (V_2, E_2) risultino connessi, dove
>   · $E_1 :=$ insieme degli archi in E i cui estremi sono in V_1,
>   · $E_2 :=$ insieme degli archi in E i cui estremi sono in V_2;
>   - sia e un arco di peso minimo (rispetto alla funzione peso w) che attraversa il taglio (V_1, V_2);
> $E_1 :=$ Quick_Something(V_1, E_1);
> $E_2 :=$ Quick_Something(V_2, E_2);
>   **endif**
>   **return** $E_1 \cup E_2 \cup \{e\}$;
> **end** Quick_Something;

(a) Si dimostri che la procedura Quick_Something(V, E) calcola un albero $\mathcal{T}$ di copertura di (V, E).

(b) $\mathcal{T}$ è necessariamente un minimo albero di copertura per (V, E) ? NO



✓ O

$|V| = n$

$(V_1, E_1) \xrightarrow{\text{per ipotesi induttiva}}$ (V_1, E_1) è un albero ricoprente

$(V_2, E_2) \xrightarrow{\text{per ipotesi induttiva}}$ (V_2, E_2) è un albero ricoprente

$\Rightarrow (V_1 \cup V_2, E_1 \cup E_2 \cup \{e\})$

$= (V, E_1 \cup E_2 \cup \{e\})$ è un albero ricoprente

ESERCIZIO 1

Siano T_1 e T_2 due MST distinti di un dato grafo non-orientato connesso e pesato (G, w). Si verifichi che:

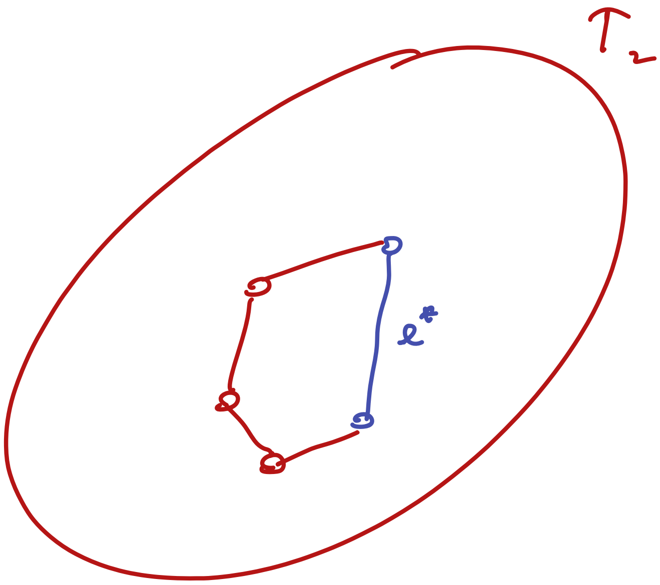
(a) $\min\limits_{e \in T_1} w(e) = \min\limits_{e' \in T_2} w(e')$,

(b) (**facoltativo**) $\max\limits_{e \in T_1} w(e) = \max\limits_{e' \in T_2} w(e')$.

$$\min_{e \in T_1} w(e) \not< \min_{e' \in T_2} w(e')$$

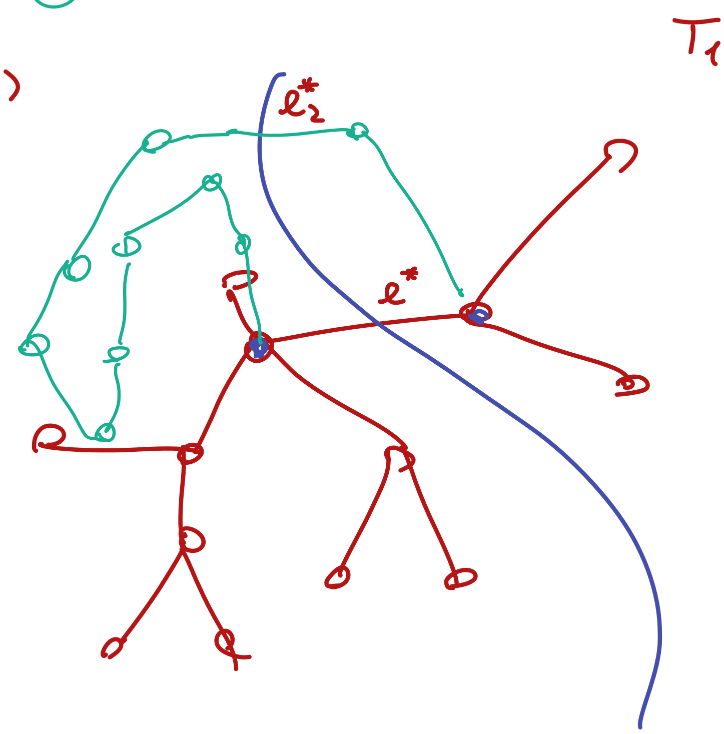
$$\not>$$

$$e^* : w(e^*) = \min_{e \in T_1} w(e)$$



$$\max_{e \in T_1} w(e) > \max_{e' \in T_2} w(e')$$

$$w(e^*) = \max_{e \in T_1} w(e)$$



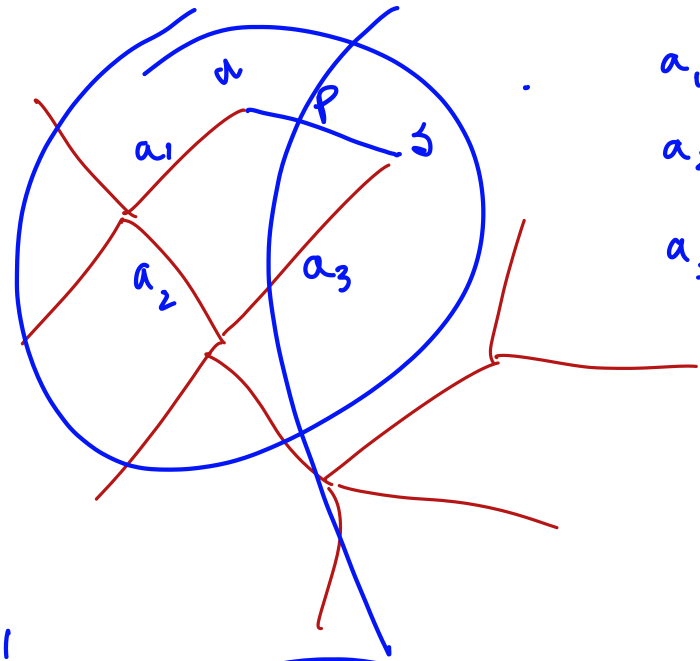
$$w(e^*) > w(e_2^*)$$

$$(T_1 \setminus \{e^*\}) \cup \{e_2^*\}$$

ESERCIZIO 1

Sia $G = (V, E)$ un grafo non orientato connesso e sia $w : E \longrightarrow \mathbb{R}$ una funzione peso su G. Sia inoltre $e \in E$ un arco di G.

Si descriva un algoritmo per stabilire se l'arco e è contenuto in qualche MST di (G, w) e se ne valuti la complessità computazionale.



$a_1 < p$

$a_2 < p$

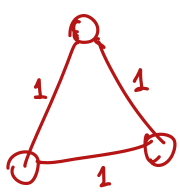
$a_3 < p$

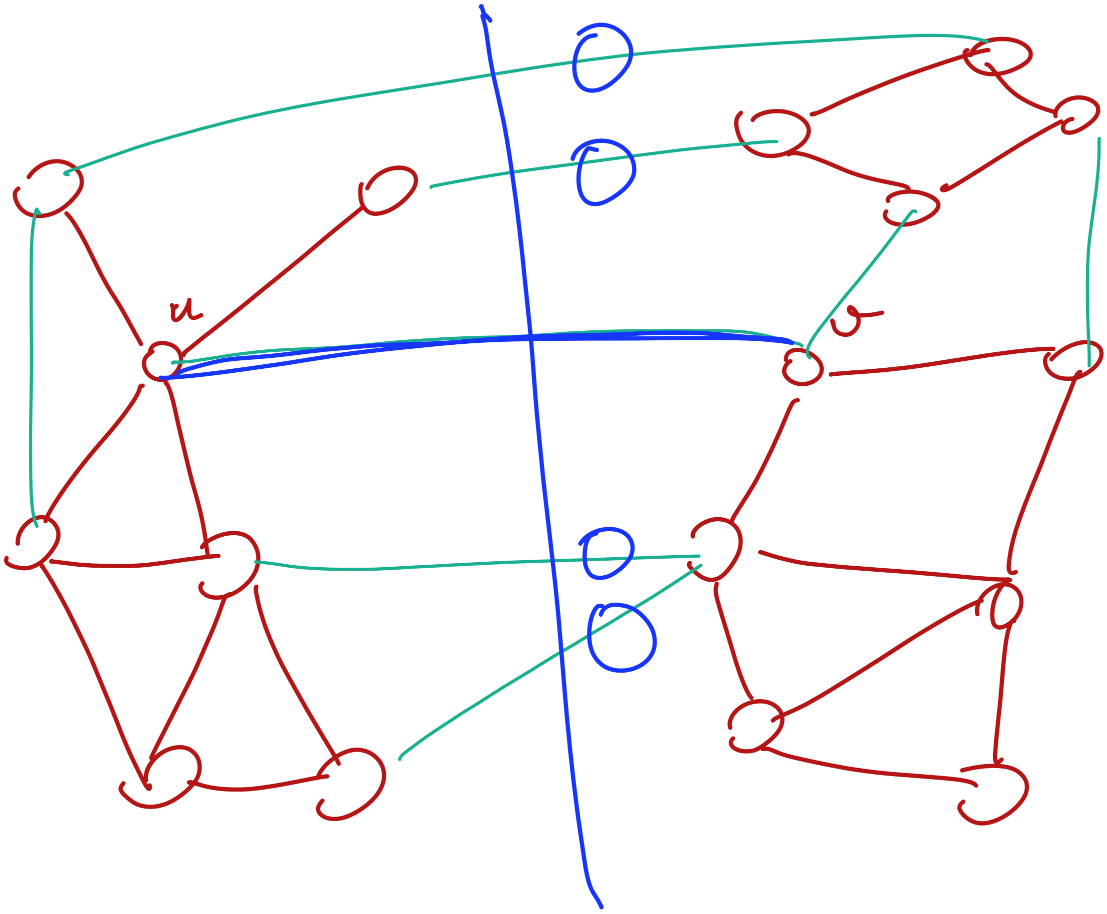
FATTO!

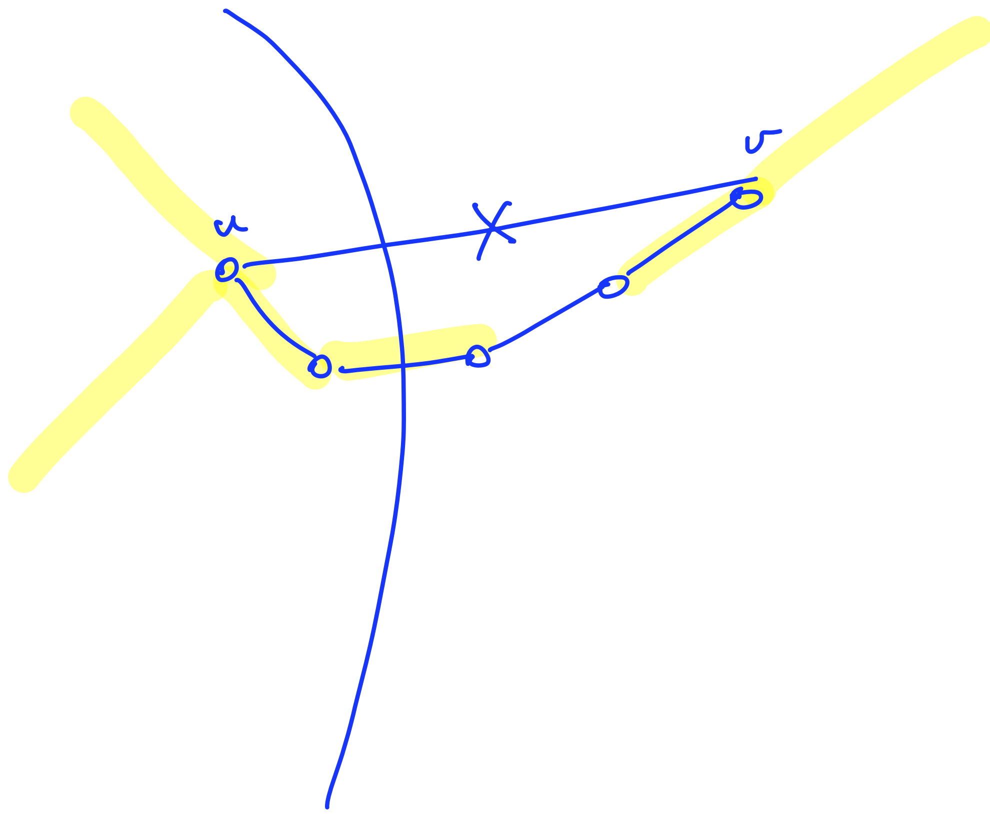
condizione necessaria e sufficiente affinché un arco (u,v) di peso p possa fare parte di un MST è che non ci sia alcun cammino in G da u a v i cui archi abbiano tutti peso strettamente minore di p

Salve prof., alla fine della lezione mi è sorta una domanda riguardante l'ultimo esercizio (quello di vedere se l'arco "e" appartiene ad un mst del grafo G): una possibile soluzione potrebbe essere quella di applicare PRIM due volte sui 2 nodi che sono collegati proprio dall'arco "e" e vedere se in uno dei due MST trovati da PRIM è presente l'arco "e"?

CONTROESEMPIO




u
v


u
v

ESERCIZIO 2

Sia $G = (V, E)$ un grafo non orientato connesso e con funzione peso $w : E \to \mathbb{R}$, e siano

$$(u_1, v_1),\ (u_2, v_2),\ \ldots,\ (u_k, v_k) \tag{*}$$

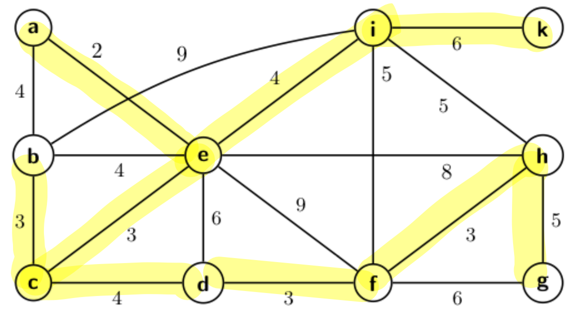
tutti e soli gli archi di G di peso minimo.

Si enunci e si dimostri una proprietà necessaria e sufficiente affinché esista un *minimum spanning tree* di G che contenga tutti gli archi (*). Quindi si proponga un algoritmo per verificare tale proprietà e se ne valuti la complessità computazionale.

ESERCIZIO 2 **(Minimum spanning trees)**

(a) Si descriva l'algoritmo di Kruskal dimostrandone anche la correttezza.

(b) Si applichino l'algoritmo di Kruskal e l'algoritmo di Prim al grafo a lato.



PRIM (a)

a	0									
b	+∞	(a,b)/4	(a,b)/4	(c,b)/3	—	—	—	—	—	—
c	+∞	+∞	(e,c)/3	—	—	—	—	—	—	—
d	+∞	+∞	(e,d)/6	(c,d)/4	(c,d)/4	—	—	—	—	—
e	+∞	(a,e)/2	—	—	—	—	—	—	—	—
f	+∞	+∞	(e,f)/9	(e,f)/9	(e,f)/9	(d,f)/3	—	—	—	—
g	+∞	+∞	+∞	+∞	+∞	+∞	(f,g)/6	(h,g)/5	(h,g)/5	—
h	+∞	+∞	(e,h)/8	(e,h)/8	(e,h)/8	(e,h)/8	(f,h)/3	—	—	—
i	+∞	+∞	(e,i)/4	(e,i)/4	(e,i)/4	(e,i)/4	(e,i)/4	(e,i)/4	—	—
k	+∞	+∞	+∞	+∞	+∞	+∞	+∞	+∞	(i,k)/6	(i,k)/6

KRUSKAL

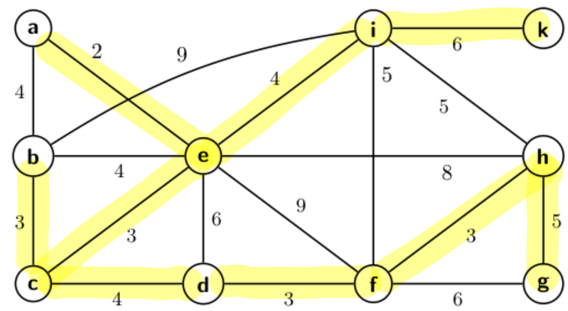
(a,e), (b,c), (c,e), (d,f), (f,h), (a,b), (b,e), (c,d), (e,i), (f,i), (h,g), (h,i), (d,e), (f,g), (i,k), (e,h), (b,i), (e,f)

(a,e), (b,c), (c,e), (d,f), (f,h), (a,b), (b,e), (c,d), (e,i), (f,i), (h,g), (h,i), (d,e), (f,g), (i,k), (e,h), (b,i), (e,f)

ESERCIZIO 2 (Minimum spanning trees)

(a) Si descriva l'algoritmo di Kruskal dimostrandone anche la correttezza.

(b) Si applichino l'algoritmo di Kruskal e l'algoritmo di Prim al grafo a lato.



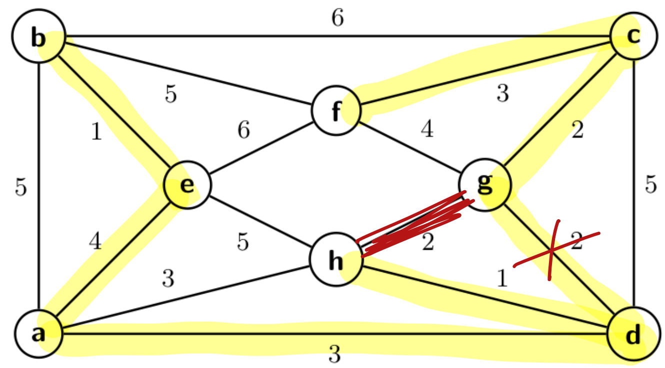
a	0	–	–	–	–	–	–	–	–	–
b	∞	(a,b)/4	(a,b)/4	(c,b)/3	–	–	–	–	–	–
c	∞	∞	(e,c)/3	–	–	–	–	–	–	–
d	∞	∞	(e,d)/6	(c,d)/4	(c,d)/4	–	–	–	–	–
e	∞	(a,e)/2	–	–	–	–	–	–	–	–
f	∞	∞	(e,f)/9	(e,f)/9	(e,f)/9	(d,f)/3	–	–	–	–
g	∞	∞	∞	∞	∞	∞	(f,g)/6	(h,g)/5	(h,g)/5	–
h	∞	∞	(e,h)/8	(e,h)/8	(e,h)/8	(e,h)/8	(f,h)/3	–	–	–
i	∞	∞	(e,i)/4	(e,i)/4	(e,i)/4	(e,i)/4	(e,i)/4	(e,i)/4	–	–
k	∞	∞	∞	∞	∞	∞	∞	∞	(i,k)/6	(i,k)/6

ESERCIZIO 2 (Alberi ricoprenti minimi)

(a) Si descriva l'algoritmo di Kruskal e lo si applichi al grafo a lato.

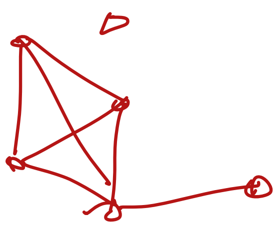
(b) Ci sono altri alberi ricoprenti minimi? SI

(c) Sia $G = (V, E)$ un grafo non orientato, pesato e connesso, e sia $w : E \to \mathbb{N}$ una funzione peso per G. Si supponga che G contenga *esattamente* 7 archi di peso unitario e nessun arco di peso strettamente inferiore ad 1. Quanti archi di peso unitario dovrà necessariamente contenere un albero ricoprente minimo per G? Perché?



(b,e), (d,h), (c,g), (d,g), (g,h), (a,d), (a,h), (c,f), (a,e), (f,g),

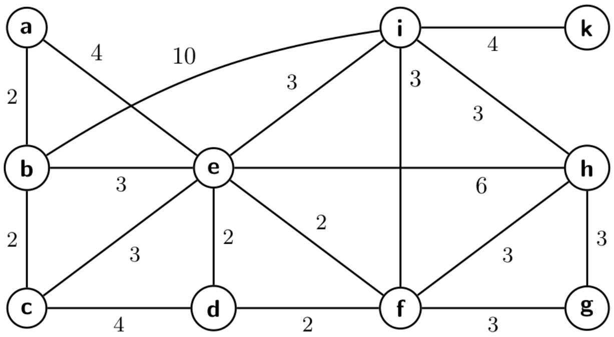
(a,b), (b,f), (c,d), (e,h), (b,c), (e,f)

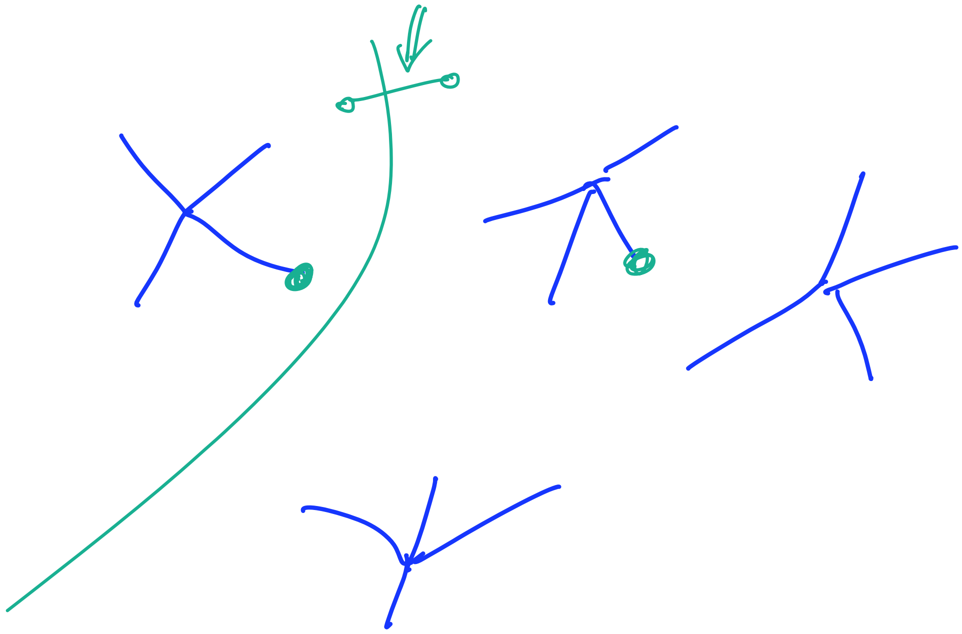


ESERCIZIO 4 (Minimum spanning trees)

(a) Si descriva l'algoritmo di Kruskal, fornendone anche lo pseudo-codice, e lo si applichi al grafo a lato.

(b) Si descrivano i "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree* e si enunci l'*invariante del colore*.
Quindi si dimostri che se dopo un certo numero di passi di colorazione il sottografo degli archi blu non forma ancora uno spanning tree, allora è possibile eseguire un passo blu.





ESERCIZIO 3

Si illustri l'algoritmo di Kruskal e se ne dimostri la correttezza.

ESERCIZIO 4 (Minimum spanning trees)

Sia $G = (V, E)$ un grafo non orientato e connesso, e sia $w\colon E \to \mathbb{R}$ una funzione peso *iniettiva* su G.
Si dimostri che G ha un unico minimum spanning tree.

ESERCIZIO 4

(a) Si descriva l'algoritmo di Prim, se ne valuti la complessità computazionale e se ne dimostri la correttezza.

(b) Sia $G = (V, E)$ un grafo non orientato connesso con funzione peso $w : E \to \mathbb{N}$ e sia E_M l'insieme degli archi di un suo albero di copertura minimo. Si supponga inoltre che G contenga esattamente tre archi distinti e_1, e_2 ed e_3 aventi peso 0. Ovviamente vale $0 \leq |E_M \cap \{e_1, e_2, e_3\}| \leq 3$.
Per ciascun valore di $k = 0, 1, 2, 3$, si determini una proprietà $\mathcal{P}_k$ di ([redacted] e_1, e_2, e_3) tale che

$$|E_M \cap \{e_1, e_2, e_3\}| = k \quad \longleftrightarrow \quad \mathcal{P}_k([redacted]\, e_1, e_2, e_3) = \textbf{true}\,.$$

(G, V) non orientato e connesso

$w : E \to \mathbb{N}$,

E_M $\quad e_1, e_2, e_3$ di peso 0

$$0 \leq |E_M \cap \{e_1, e_2, e_3\}| \leq 3$$

$|E_M \cap \{e_1, e_2, e_3\}| = 0 \longleftrightarrow P_0 = \perp$

$|E_M \cap \{e_1, e_2, e_3\}| = 1 \longleftrightarrow P_1 = \perp$

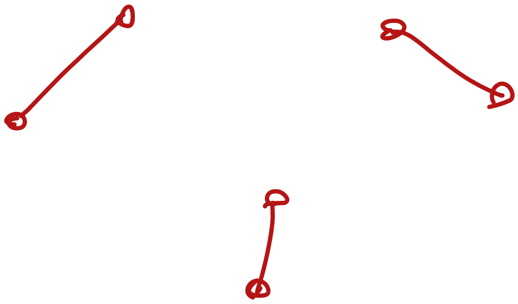
$|E_M \cap \{e_1, e_2, e_3\}| = 2 \longleftrightarrow e_1, e_2, e_3$ formano un ciclo

$|E_M \cap \{e_1, e_2, e_3\}| = 3 \longleftrightarrow e_1, e_2, e_3$ non formano un ciclo

ESERCIZIO 4

Sia dato un grafo connesso non orientato $G = (V, E)$ con funzione peso $w : E \longrightarrow \mathbb{R}$. Sia inoltre F un sottoinsieme aciclico di E. Un F-*spanning tree* di G è uno spanning tree di G il cui insieme di archi contiene F.

Si descriva un algoritmo efficiente (valutandone la complessità e dimostrandone la correttezza) che calcoli un minimo F-spanning tree di G.

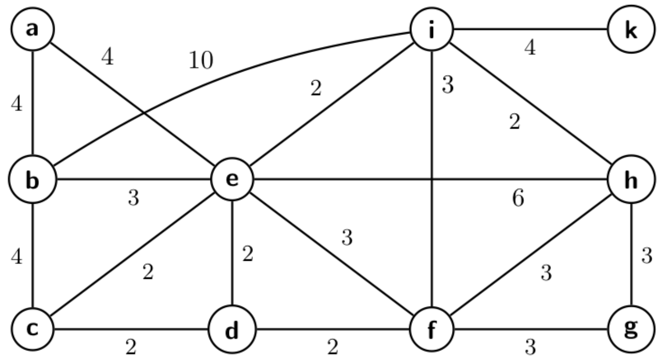


ESERCIZIO 4 (Minimum spanning trees)

(a) Si descriva l'algoritmo di Prim (anche mediante il suo pseudo-codice) e lo si applichi al grafo a lato, iniziando dal nodo **a**.

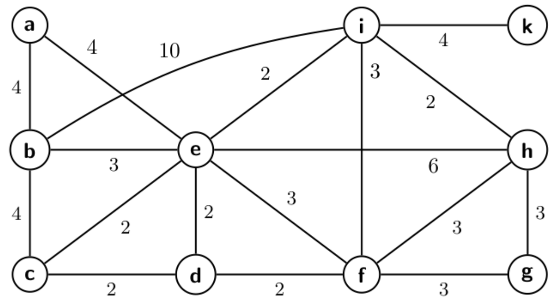
(b) Per ciascuno dei seguenti archi si stabilisca se possa fare parte di un *minimum spanning tree* del grafo a lato:
$(\mathbf{c}, \mathbf{d})$, $(\mathbf{f}, \mathbf{h})$, $(\mathbf{f}, \mathbf{g})$, $(\mathbf{e}, \mathbf{f})$.

(c) Si descrivano i cosiddetti "passi blu" e "passi rossi" neglialgoritmi per il calcolo del *minimum spanning tree*. Quindi si enunci l'*invariante del colore* e lo si dimostri limitatamente ai passi blu.



ESERCIZIO 2 (Minimum spanning trees)

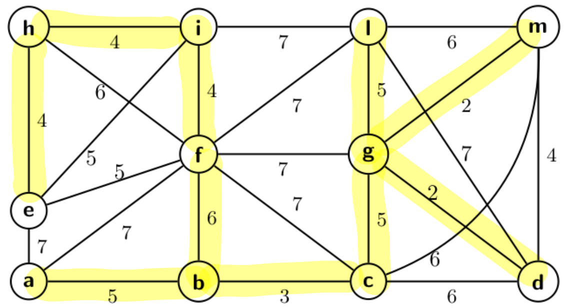
(a) Si descriva l'algoritmo di Kruskal, fornendone anche lo pseudo-codice, e lo si applichi al grafo a lato, utilizzando tra archi del medesimo peso l'ordinamento lessicografico (per cui, ad es., (a, b) precede (a, e) che, a sua volta, precede (i, k)), ove gli archi stessi sono rappresentati lessicograficamente (per cui, ad es., la rappresentazione di riferimento dell'arco tra i nodi e ed a è (a, e)).



(b) Si descrivano i "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree* e si enunci l'*invariante del colore*.
Quindi si dimostri che se dopo un certo numero di passi di colorazione il sottografo degli archi blu non forma ancora uno spanning tree, allora è possibile eseguire un passo blu.

ESERCIZIO 2 (Minimum spanning trees)

(a) Si descriva l'algoritmo di Kruskal, fornendone anche lo pseudo-codice, e lo si applichi al grafo a lato, utilizzando tra archi del medesimo peso l'ordinamento lessicografico (per cui, ad es., (a, b) precede (a, e) che, a sua volta, precede (i, k)), ove gli archi stessi sono rappresentati lessicograficamente (per cui, ad es., la rappresentazione di riferimento dell'arco tra i nodi e ed a è (a, e)).

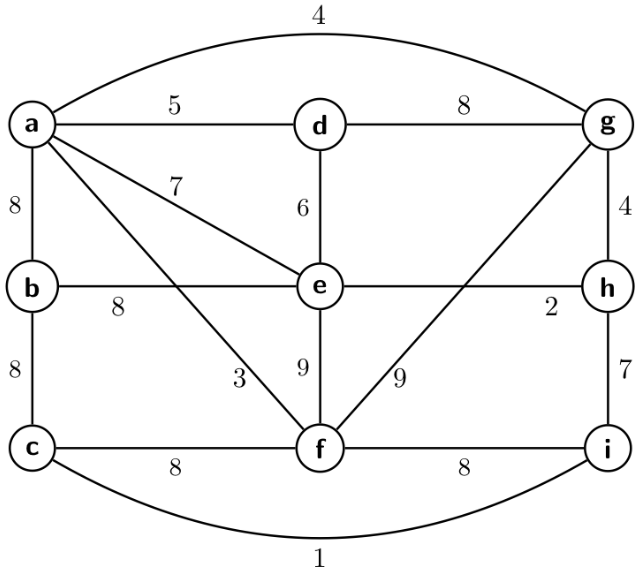


(b) Si descrivano i "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree* e si enunci l'*invariante del colore*.
Quindi si dimostri che se dopo un certo numero di passi di colorazione il sottografo degli archi blu non forma ancora uno spanning tree, allora è possibile eseguire un passo blu.

ESERCIZIO 5 (Minimum spanning trees)

(a) Si descriva l'algoritmo di Kruskal (anche mediante il suo pseudo-codice) e lo si applichi al grafo a lato.

(b) Si descrivano i cosiddetti "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree*. Quindi si enunci e si dimostri l'*invariante del colore*.

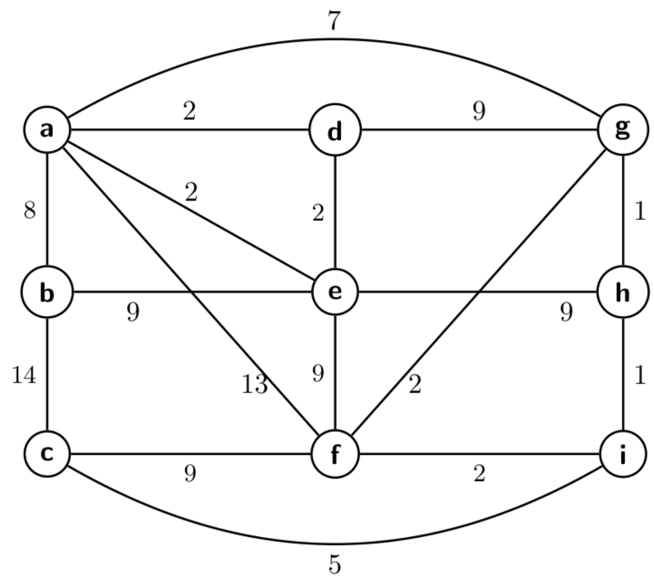


ESERCIZIO 5 (Minimum spanning trees)

(a) Si descriva l'algoritmo di Kruskal (anche mediante il suo pseudo-codice) e lo si applichi al grafo a lato.

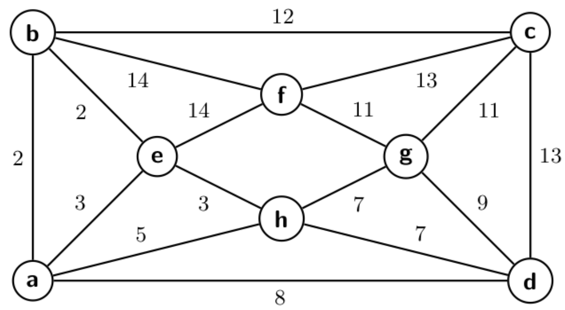
(b) Quanti alberi ricoprenti minimi ha il grafo a lato?

(c) Si descrivano i cosiddetti "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree*. Quindi si enunci e si dimostri l'*invariante del colore*.



ESERCIZIO 1 (Minimum spanning tree)

1. Si descriva l'algoritmo di Kruskal e lo si applichi al grafo a lato.
2. Ci sono altri alberi ricoprenti minimi? (Giustificare la risposta.)
3. Si descrivano i cosiddetti "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree* nei grafi non-orientati, connessi e pesati. Quindi si enunci l'*invariante del colore* e lo si dimostri limitatamente ai passi blu.



ESERCIZIO 1 (Minimum spanning tree)

1. Si descriva l'algoritmo di Prim e lo si applichi al grafo a lato a partire dal vertice **g**.
2. Si descrivano i cosiddetti "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree* nei grafi non-orientati, connessi e pesati. Quindi si enunci l'*invariante del colore* e lo si dimostri limitatamente ai passi rossi.
3. Dimostrare che non esiste alcun albero ricoprente minimo del grafo a lato contenente l'arco $(\mathbf{a}, \mathbf{d})$.