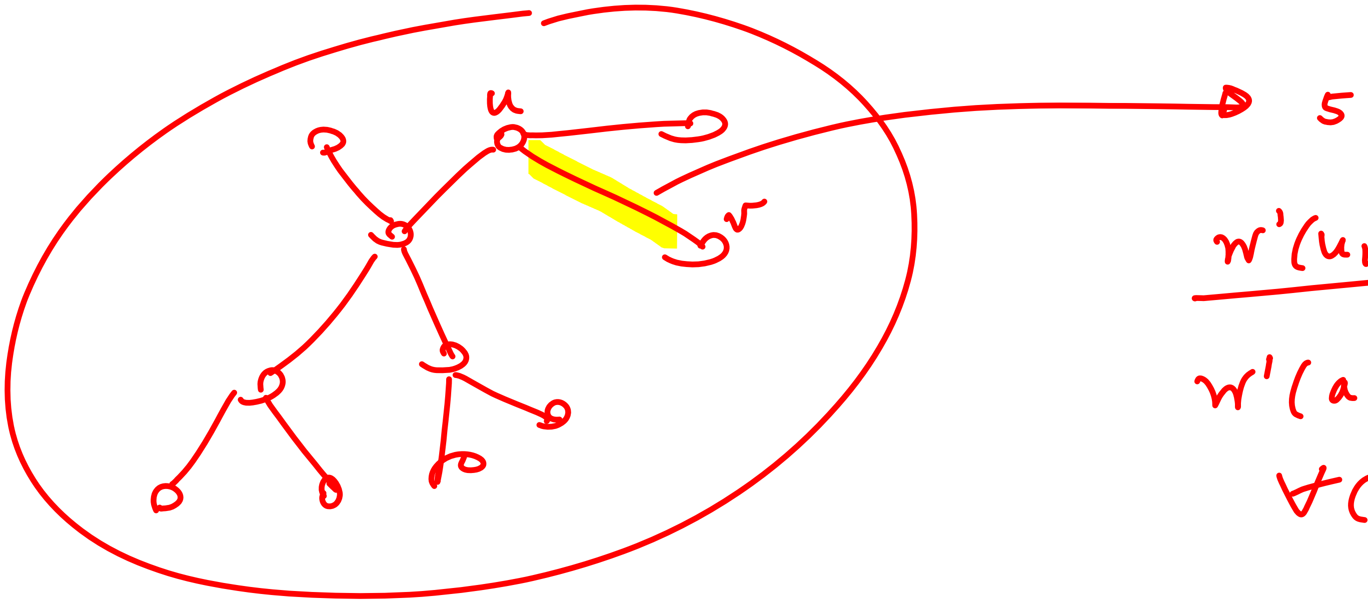
ESERCIZIO 4

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.

$F \subseteq E$, F aciclico

F-spanning tree : spanning tree T tale che $F \subseteq T$



$w'(u,v) < 5$

$w'(a,b) = w(a,b)$

$\forall (a,b) \neq (u,v)$

$F \subseteq T$ Tesi: T è un Minimum F-spanning tree.

T_1 minimo F-spanning tree $\quad w(T_1) < w(T)$

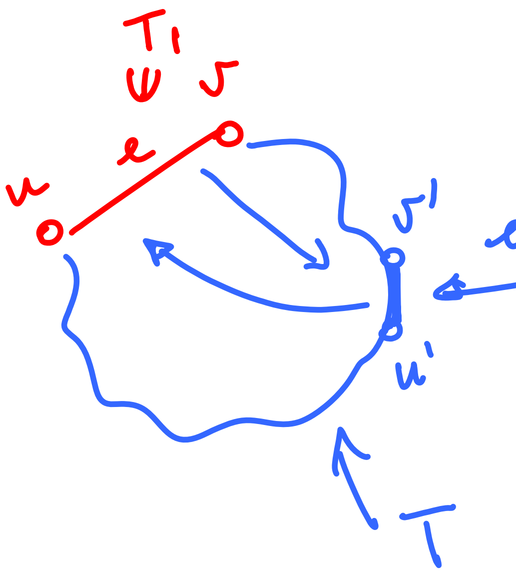
$\boxed{e \in T_1 \Delta T}$ di peso minimo $\quad |T_1' \Delta T| < |T_1 \Delta T|$

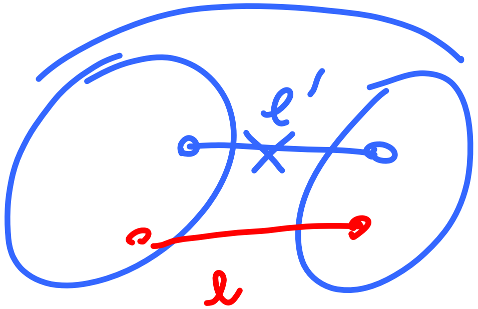
$e \notin F \quad \boxed{w(e) < w(e')}$

$e \in T_1 \setminus T$, oppure $e \in T \setminus T_1$

$T \setminus \{e'\} \quad \boxed{w(e) \leq w(e')}$

$w(e) \neq w(e')$

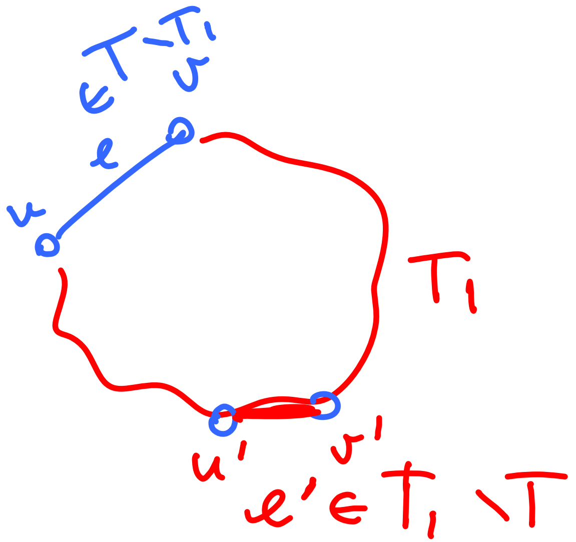




$T_1' = (T_1 \setminus \{e\}) \cup \{e'\}$

$|T_1' \Delta T| < |T_1 \Delta T|$

$e \in T \setminus T_1$



$w(e) \leq w(e')$

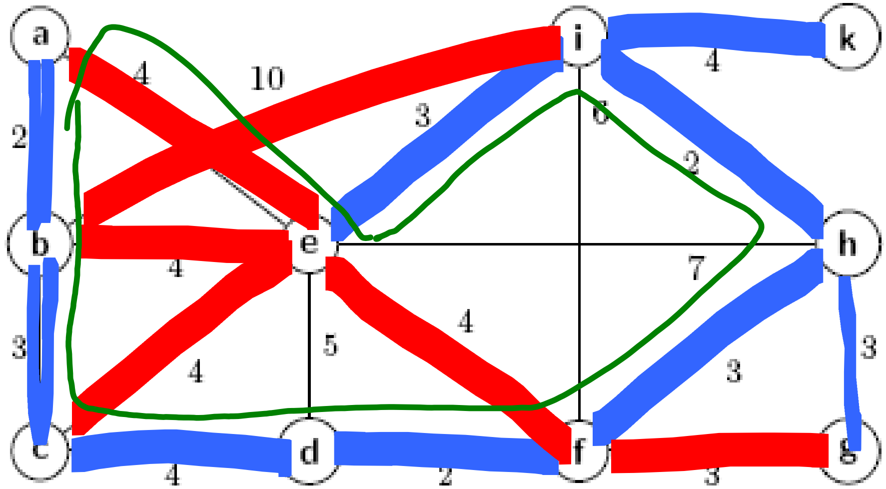
$w(e) = w(e')$

$w(e) < w(e')$

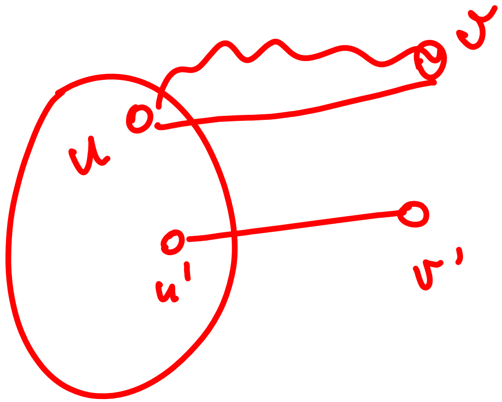
ESERCIZIO 4 (Minimum spanning trees)

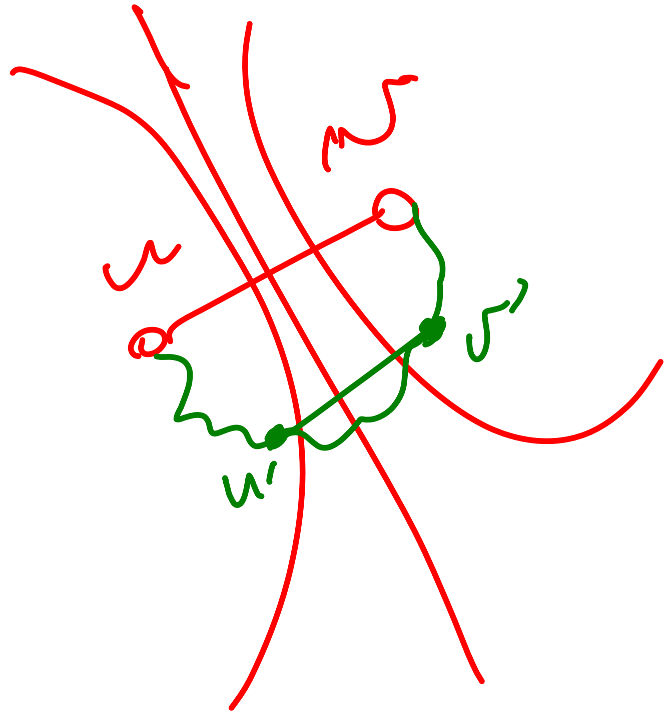
(a) Si descriva l'algoritmo di Kruskal (anche mediante il suo pseudo-codice) e lo si applichi al grafo a lato.

(b) Si descrivano i cosiddetti "passi blu" e "passi rossi" negli algoritmi per il calcolo del *minimum spanning tree* nei grafi non-orientati, connessi e pesati. Quindi si enunci l'*invariante del colore* e lo si dimostri limitatamente ai passi rossi.



$\boxed{a, e, i, h, f, d, c, b, a}$



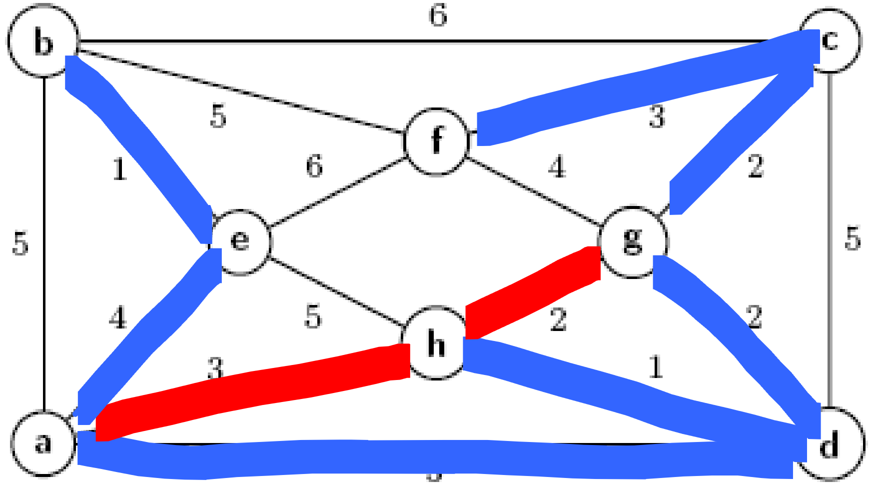


ESERCIZIO 2 (Alberi ricoprenti minimi)

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

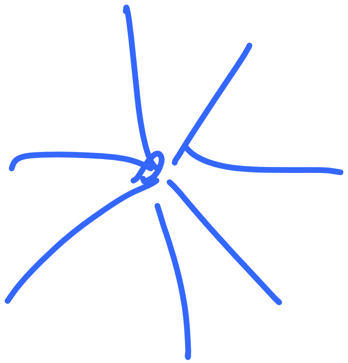
(b) Ci sono altri alberi ricoprenti minimi?

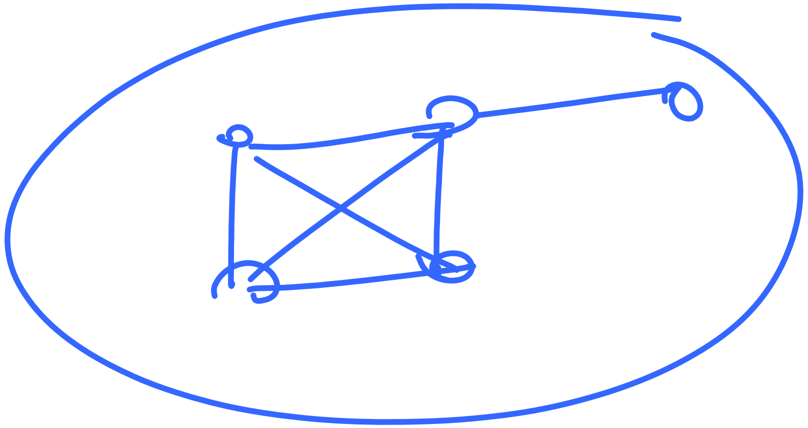
(c) Sia $G = (V, E)$ un grafo non orientato, pesato e connesso, e sia $w : E \to \mathbb{N}$ una funzione peso per G. Si supponga che G contenga *esattamente* 7 archi di peso unitario e nessun arco di peso strettamente inferiore ad 1. Quanti archi di peso unitario dovrà necessariamente contenere un albero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

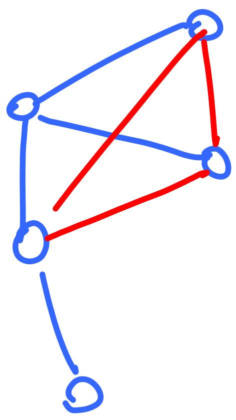
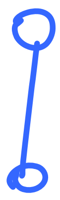


$(h,g) \rightleftarrows (g,d)$ 2

$(a,h) \rightleftarrows (a,d)$ $\frac{2}{4}$





$$\underbrace{5 \leq N \leq 14}_{k \text{ componenti}}$$

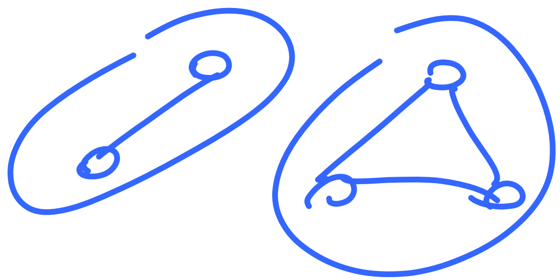
$$N-k>3$$

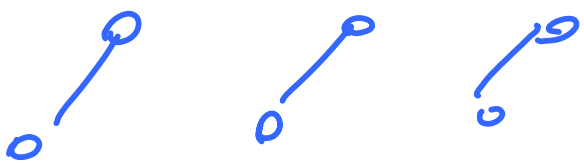
$$n_1 + n_2 + \dots + n_k = N \quad \Leftarrow$$

$$(n_1-1) + (n_2-1) + \dots + (n_k-1) = N-k = 3$$

1
sola
componente

$$k = N-3$$





N	k
5	2
$\boxed{6}$	$\boxed{3}$
~~7~~	~~4~~
~~8~~	~~5~~
⋮	⋮
~~14~~	~~11~~

ESERCIZIO 2

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

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

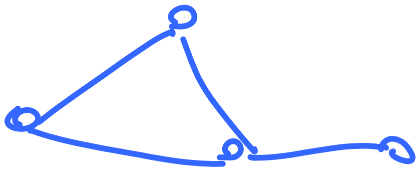
tutti e soli gli archi di G di peso minimo.

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.

9 archi

$\boxed{7}$ archi

$\boxed{4}$ archi



1

1 1

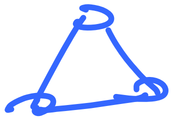
1 2 1 +1

1 3 3 1

1 4 6 4 1

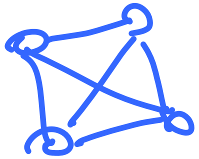
1 5 10 10 5 1

1 6 15 20 15 6 1



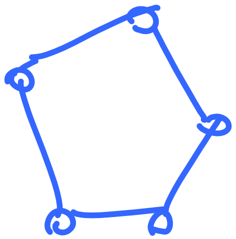
3

$3+1 = 4$



6

$6+1 = 7$

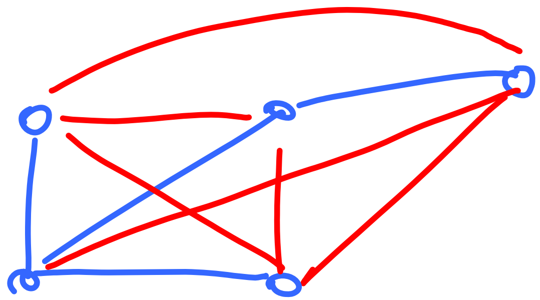


$$\binom{5}{2} = \frac{5!}{2!\,3!} = \frac{\cancel{4}^{2}\cdot 5}{\cancel{2}} = \boxed{10}$$

$10+1 = 11$

$\underbrace{7\ \ 8\ \ 9\ \ 10}_{4}$

$\underbrace{11\ \ 12\ \ 13\ \ 14\ \ 15}_{5}$



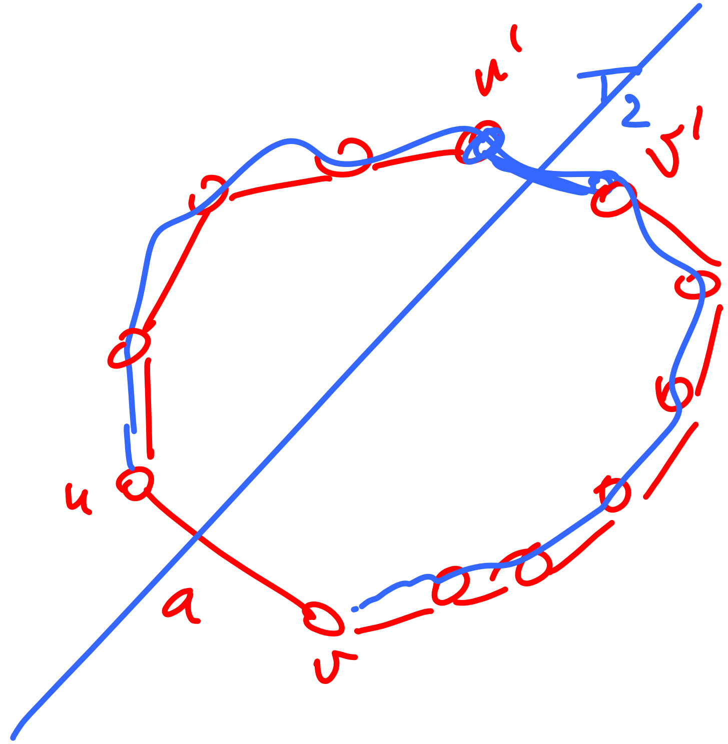
ESERCIZIO 3

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

(a) $\overbrace{\min_{e \in T_1} w(e)}^{a} = \overbrace{\min_{e' \in T_2} w(e')}^{b}$,

(b) $\max_{e \in T_1} w(e) = \max_{e' \in T_2} w(e')$.

$a > b$



$(u,v) \in T_1$