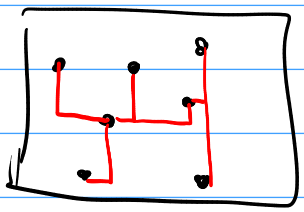
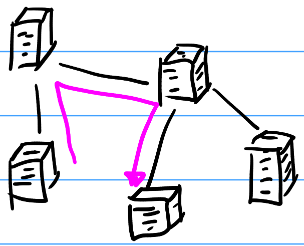
ALBERI DI COPERTURA ... MINIMI (cap 23)



• abbiamo dei contatti in un circuito che vogliamo mettere sulla stessa linea collegandoli tutti tra loro.



Abbiamo delle centraline (e.g. telefoniche) e dobbiamo fare in modo che ogni coppia di centraline nel sistema possa comunicare

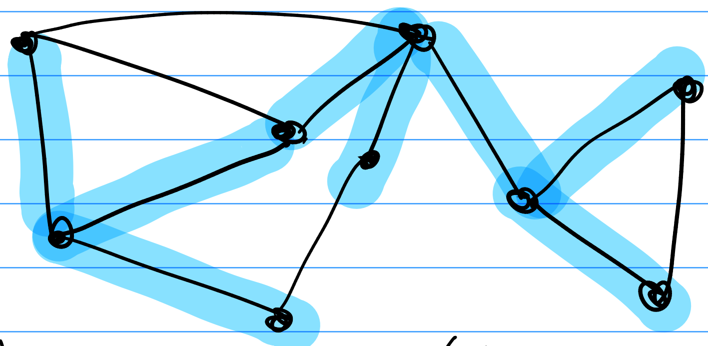
Dato un GRAFO SEMPLICE $G=(V,E)$ un

- ALBERO DI COPERTURA
- ALBERO DI CONNESSIONE (MAI SENTITO MA IL LIBRO LI CHIAMA COSÌ)
- SPANNING TREE (IN INGLESE)

è un sottoinsieme di archi $T \subseteq E$ tale che

- $G=(V,T)$ ha una singola componente connessa (OVVERO OGNI COPPIA DI VERTICI È COLLEGATA DA UN CAMMINO)
- T è minimale (OVVERO ELIMINANDO UN ARCO SI SCONNETTE IL GRAFO)

E.g.



Facciamo diverse osservazioni (lasciate per esercizio)

① Solo un grafo connesso può avere un albero di copertura

② Un albero di copertura non contiene cicli (e quindi è corretto chiamarlo albero)

3) Qualunque albero di copertura ha un vertice di grado 1

3bis) Qualunque albero di copertura di n vertici ha esattamente (n-1) archi

3tris) Qualunque grafo connesso di n vertici e (n-1) archi è un albero

4) In un albero di copertura, per ogni due vertici esiste un UNICO CAMMINO che li connette

INDIZI

1) rileggete le definizioni

2) Verificate che se $G=(V,E)$ è connesso e contiene un ciclo, è possibile eliminare un arco dal ciclo senza disconnettenere nessuna coppia di vertici del grafo

3) Si parte da un vertice e si produca un cammino più lungo possibile fino a che
- si torna su un vertice già visitato (CICLO!)
- si arriva ad un vertice da cui non si può uscire (DEGREE ≤ 1)

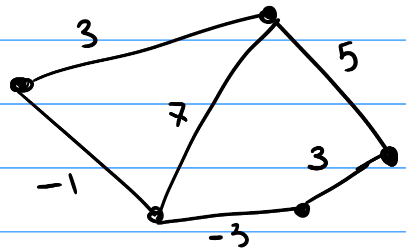
3bis) Per induzione $n=1$, 0 archi
$n>1$, trovate un vertice di grado 1. Eliminate il vertice e un arco. Usate l'ipotesi induttiva

4) Mostrate che due cammini distinti producono un ciclo

ALBERI DI COPERTURA ... MINIMI?

In questo e in altri casi è possibile avere dei PESI/COSTI sugli archi di un grafo.

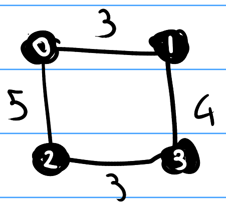
- ARCHI PESATI



- E.g. il peso di un arco può rappresentare
  - LA DISTANZA TRA I NODI
  - UN COSTO / GUADAGNO DI UNA CONNESSIONE

GRAFI PESATI

- In un grafo pesato $G=(V,E,w)$ $w: E \to \mathbb{R}$ è una funzione che assegna un peso ad ogni arco



LISTE

0 → (1,3) → (2,5)
1 → (0,3) → (3,4)
2 → (0,5) → (3,3)
3 → (1,4) → (2,3)

MATRICI DI ADIACENZA

	0	1	2	3
0	–	3	5	–
1	3	–	–	4
2	5	–	–	3
3	–	4	3	–

- oppure potete tenere i pesi in un dizionario a parte (così la lista di adiacenza può essere usata anche per algoritmi che ignorano i pesi)

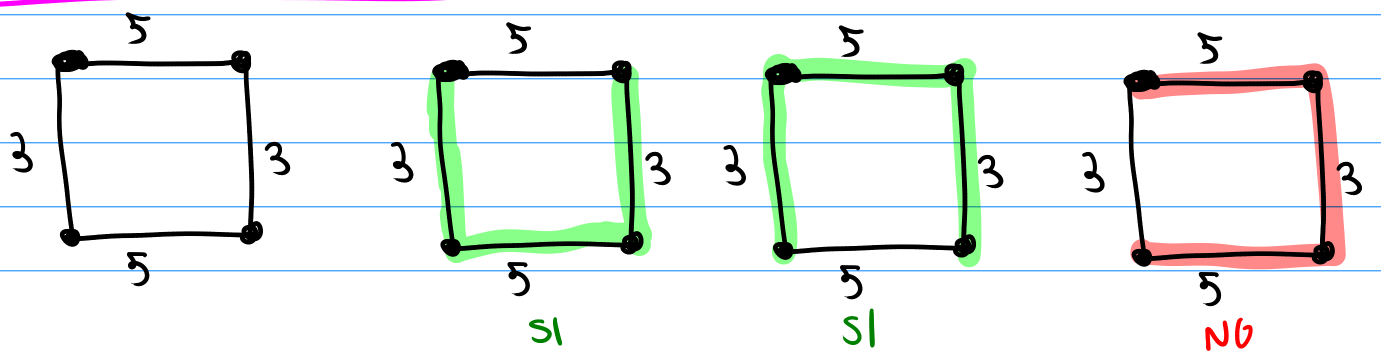
Def $G=(V,E,w)$ grafo semplice, connesso, pesato

- $\text{costo}(E') = \sum_{e \in E'} w(e)$ per $E' \subseteq E$

Se T è un ALBERO DI COPERTURA di G, allora costo(T) è il "costo dell' ALBERO DI COPERTURA T"

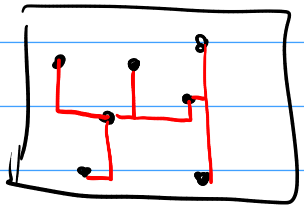
Problema Data $G=(V,E,w)$ connesso, semplice, pesato si vuole trovare un albero di copertura T il cui costo è il minore possibile

E.g.

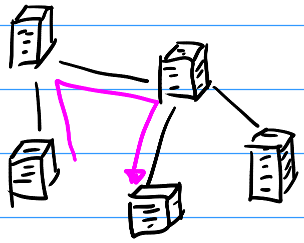


Il problema di trovare un albero di copertura minimo

è interessante perché nelle applicazioni il costo dei collegamenti ha corrispondenze naturali nei problemi reali



- il costo delle piste delle tracce dei collegamenti



- il costo delle linee di connessione

- Vediamo due algoritmi GREEDY
  - Kruskal
  - Prim

- Entrambi gli algoritmi partono con una soluzione vuota $T := \emptyset$ e ad ogni round aggiungono un arco e alla soluzione

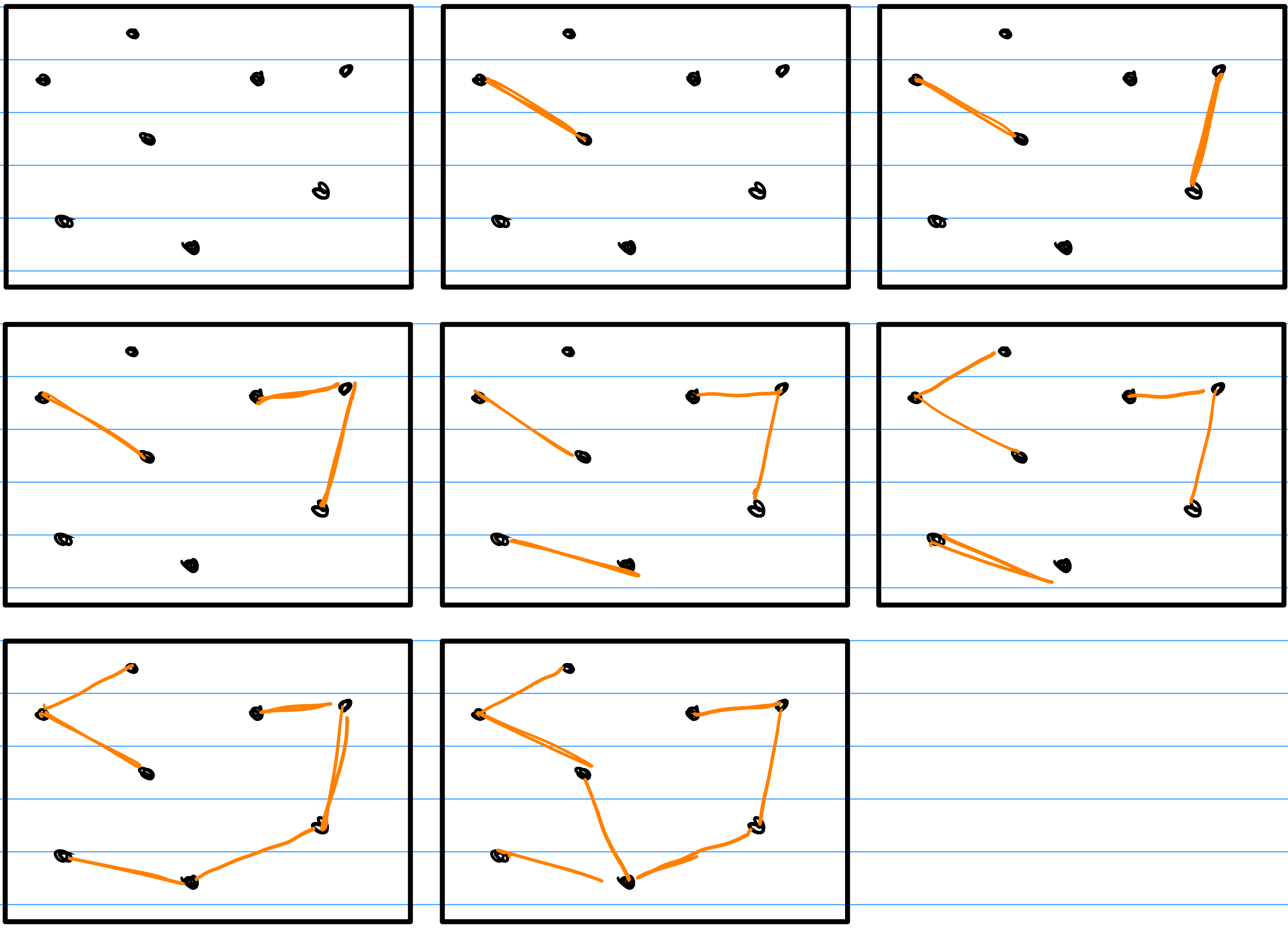
$$T := T \cup \{e\}$$ in modo tale che

T è sempre contenuto in un albero di costo minimo.

- Dopo n-1 passi (perché n-1?) T è un insieme di n-1 archi contenuto in un albero di copertura minimo, quindi ... ?

- La differenza tra i due algoritmi è nella scelta del prossimo arco da aggiungere.

Kruskal

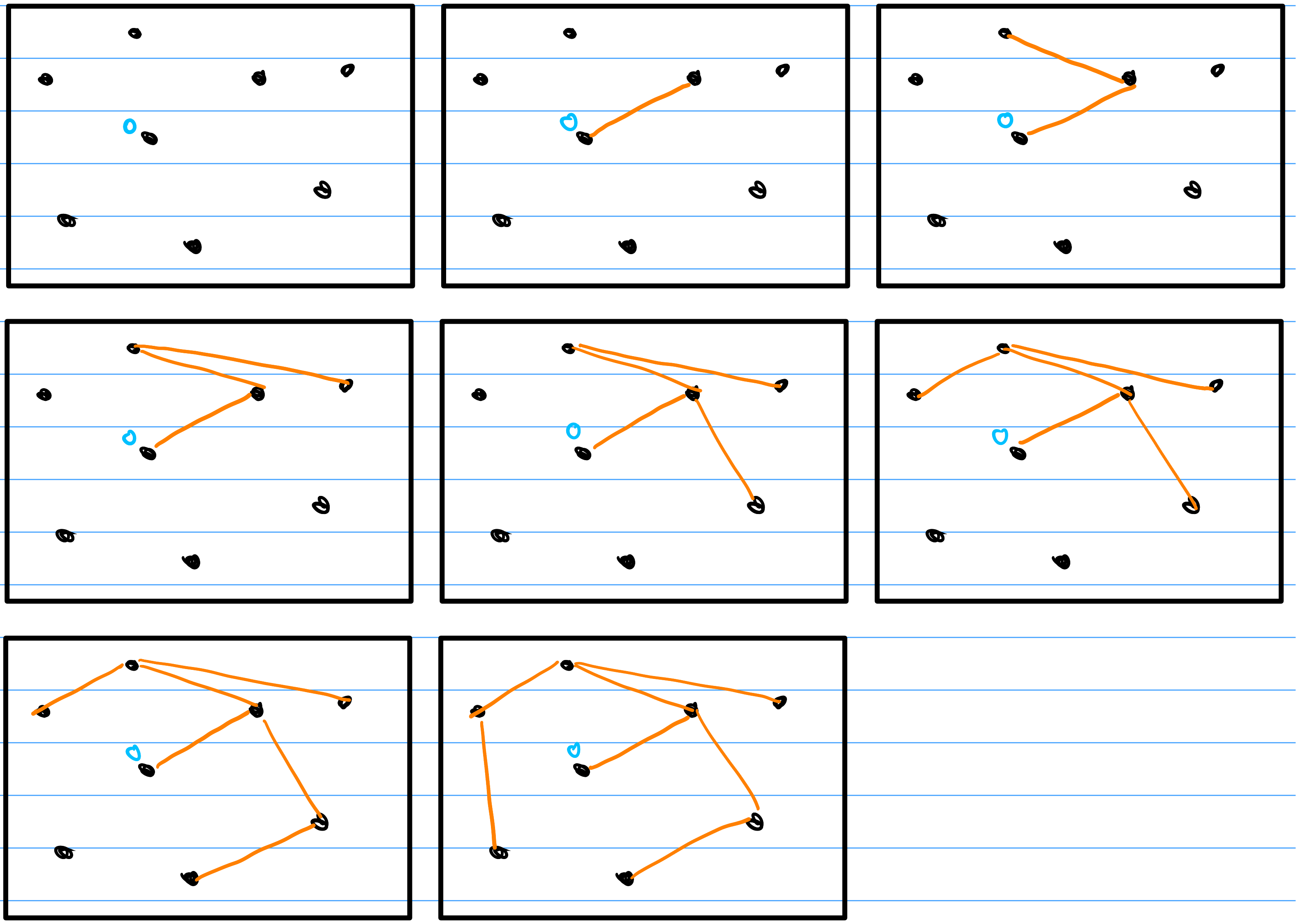


- Si parte con n componenti connesse di un vertice ognuna

STRATEGIA GREEDY

- Ad ogni passo si trova l'arco MENO COSTOSO tra quelli che UNISCONO DUE COMPONENTI CONNESSE SEPARATE
- Dopo $n-1$ passi si ha una sola componente connessa ovvero un albero di copertura

Prim



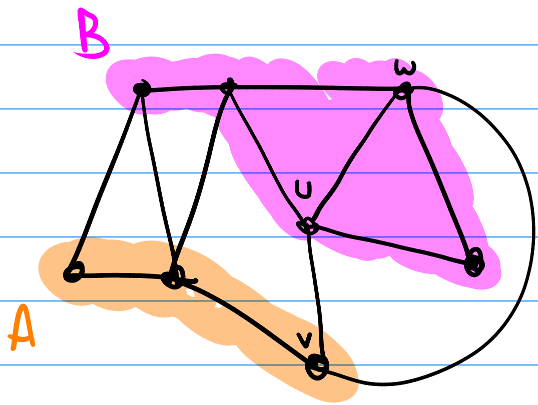
- Viene mantenuto un singolo albero T contenente inizialmente un solo vertice (ad esempio il vertice 0)
- Ad ogni passo, uno dei vertici isolati viene agganciato a T utilizzando l'arco PIÙ ECONOMICO POSSIBILE
- Dopo $n-1$ passi si ha un albero di copertura di costo minimo.

Teorema del taglio (teorema 23.1)

<u>Def</u> In un grafo $G=(V,E)$
un taglio è una partizione A,B di V

$$\left[\begin{array}{l} A \cup B = V \\ A \cap B = \emptyset \end{array}\right]$$

un arco $\{u,v\}$ attraversa la partizione A,B
oppure equivalentemente
A,B taglia l'arco $\{u,v\}$
se $|A \cap \{u,v\}| = |B \cap \{u,v\}| = 1$



- $\{u,v\}$ e $\{w,v\}$ attraversano A,B
- $\{u,w\}$ non attraversa A,B

Nomenclatura: dato $G=(V,E,w)$ un grafo semplice pesato

* $S \subseteq E$ è una <u>soluzione parziale</u> se esiste $T \subseteq E$

$S \subseteq T$ e T è un albero di copertura minimo

* Data una soluzione parziale S, un arco $\{u,v\}$ è <u>sicuro</u> per S se

$S \cup \{\{u,v\}\}$ è una soluzione parziale

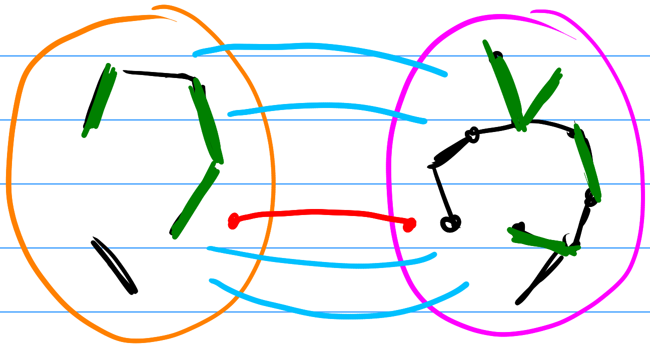
Quindi il nostro scopo (sia in Kruskal che in Prim) sarà di partire da

- soluzione parziale $\emptyset$
- Dato $\{e_1, e_2, \dots, e_\ell\}$ trovare un arco sicuro $e_{\ell+1}$

Thm Se S é una soluzione parziale

$V = A \cup B$ é un taglio tale che nessun arco di S lo attraversa

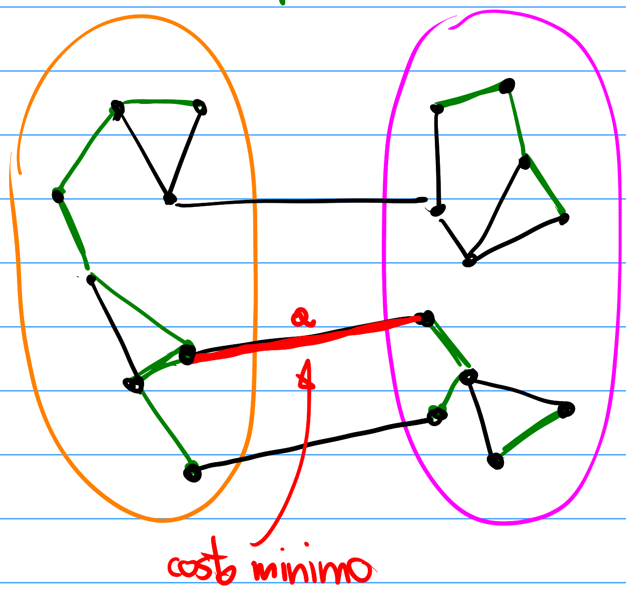
allora un arco di costo minimo tra quelli che attraversano il taglio é sicuro per S



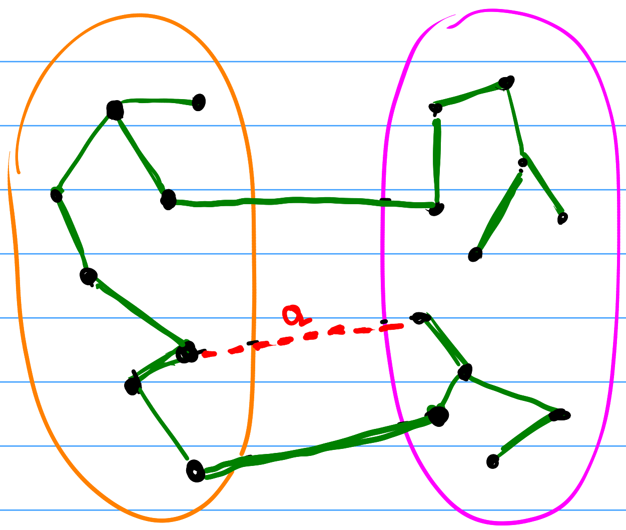
(ovvero puó essere aggiunto da S senza pregiudicare la ricerca dell'albero minimo)

dim

In verde la sol. parziale S

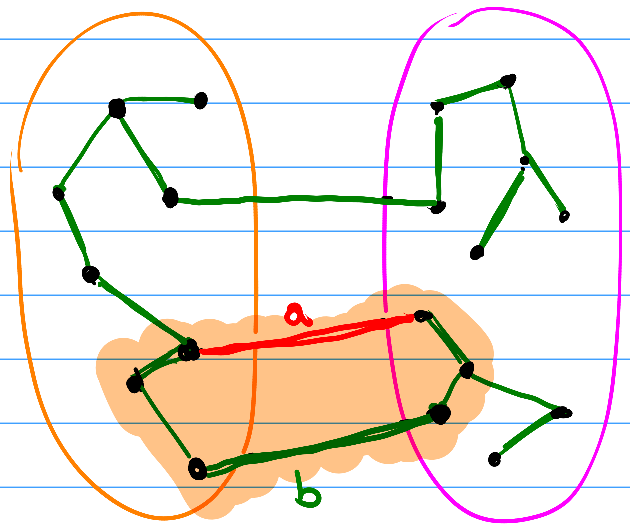


In verde la sol. T



- La soluzione finale T deve contenere degli archi che attraversano il taglio (altrimenti disconnesso)
- Se é in T, allora siamo a posto
- altrimenti $T \cup \{ \}$ ha un ciclo che coinvolge e un altro arco b di T che attraversa il taglio

9



- eliminando un arco qualunque dal ciclo, si riottiene un albero di copertura
- $w(a) \leq w(b)$
- $b \notin S$

Quindi $T' = T \cup \{a\} / \{b\}$ è
- un albero di copertura
- contiene S
- $costo(T') \leq costo(T)$

Perció a è sicuro per S $\square$