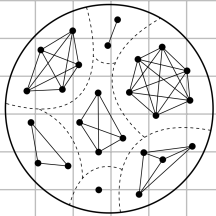


## Clustering:

Con clustering si intende la suddivisione di un grafo in più cluster (sottografo).

Es:



Noi abbiamo due obiettivi:

- Clustering Ottimale: avere un raggruppamento ottimale dei nodi
- Massimizzare lo spazamento: con spazamento si intende l'arco di peso minimo incidente su 2 nodi di cluster diversi

Come ottenere queste informazioni?

Basta modificare leggermente l'algoritmo di Kruskal

Kruskal Clustering ( $K$ ) { //  $K$  = numero di cluster presenti

Finché i raggruppamenti sono  $+ di K$

scegli  $(u, v)$  di peso minimo rimanente

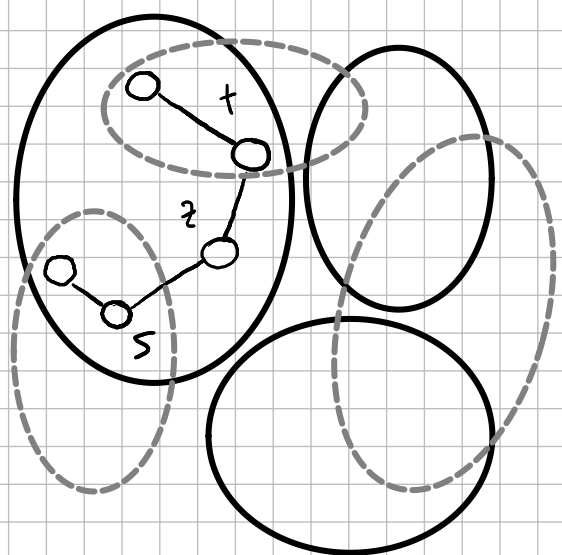
Unisci insieme  $(u)$ , insieme  $(v)$

Spazamento = per  $(t, z)$  di peso minimo rimanente tale che  $t$  e  $z$  in gruppi diversi

Da notare come lo spazamento non sia semplicemente un arco estratto dall'Hop, ma deve essere anche di gruppi (cluster) diversi. In sintesi: puoi dire che è il primo arco che Kruskal avrebbe preso in considerazione dopo il `while`.

## Dimostrazione

Dimostrare che tra tutti i raggruppamenti quello scelto da Kruskal sia quello con spazamento massimo



\_\_\_\_\_ Cluster di Kruskal (K)

----- Cluster generici (Q)

I due nodi t e s fanno parte dello stesso cluster di Kruskal ma in cluster generici diversi.

Per Kruskal sono finiti nello stesso insieme per una serie di archi che l'algoritmo ha scelto.

Quindi possiamo dire che:

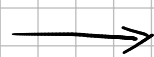
//  $sp_K$ : spazamento di Kruskal

$sp_K \geq$  peso di qualsiasi arco che

Quindi:

$$\left. \begin{array}{l} sp_k \geq \text{peso}(t, z) \\ sp_k \geq \text{peso}(z, s) \end{array} \right\} \geq sp_a$$

Quindi otteniamo



$$sp_k \geq sp_a$$

quindi Kruskal  
effettivamente

massimizza lo spazamento.

Quando fai la dimostrazione ricordati che  $u, v$  devono far parte del cluster di Kruskal.

Anche Kruskal è greedy, perché l'MST ci permette una soluzione Greedy.

Infatti:

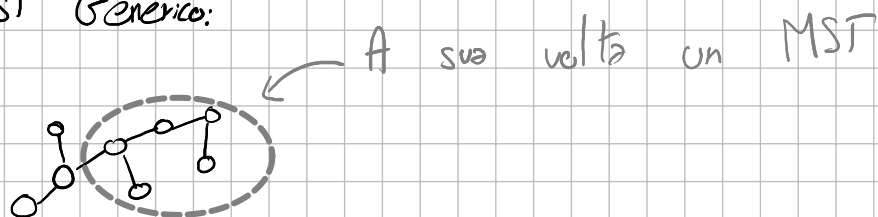
- per Dijkstra è vero perché la sotto struttura del cammino minimo è ottima

// il sottocammino di un cammino minimo è anch'esso un cammino minimo.

- per Kruskal (e prim) stessa cosa, anche in questo caso la sotto struttura è ottima.

Esempio:

MST Generico:



Un MST è fatto da tanti sottogrf. MST connessi fra di loro

Prim sfrutta questa proprietà quando esplora la frontiera

Kruskal sfrutta questa proprietà quando rimuove l'elemento dall'heap.