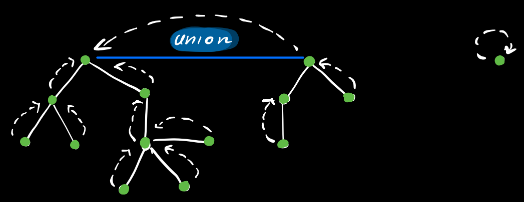
Insiemi disgiunti → rappresentati da Grafi, in cui ogni nodo punta al "nodo padre".



Le **OPERAZIONI** su INSIEMI DISGIUNTI DIVENTANO:

- **MAKESET (x)** → Albero con un nodo $\Theta(1)$
- **FINDSET (x)** → Seguire il puntatore al padre fino a trovare NIL $O(n)$
- **UNION (x, y)** → unire 2 alberi, basta definire una radice come Figlia dell'altra.

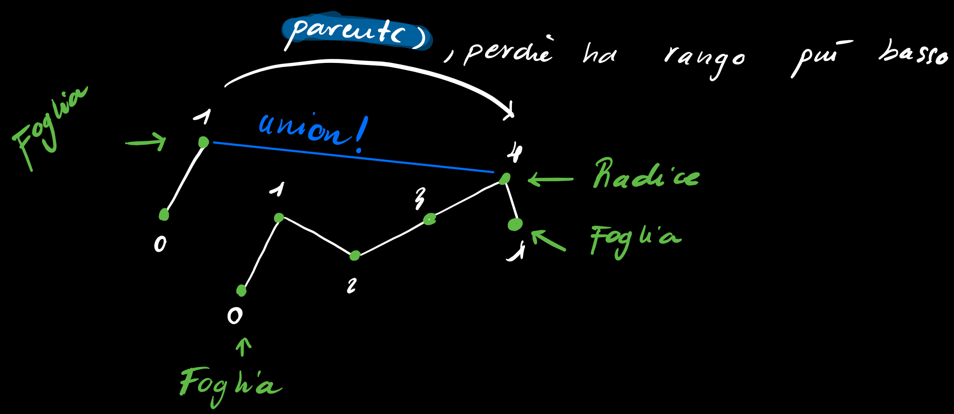
Unione per rango → risolve la situazione in cui:

Si unisce un grafo "piccolo" come rappresentante di un grafo molto più "grande".

Ripetutamente si perde la forma a "grafo" e ci si avvicina molto più ad una linked-list, PERDENDO IL VANTAGGIO DI USARE UN GRAFO.

∀nodo è associato un "rango", che corrisponde alla "distanza dalla foglia" = a sua volta al "ramo più lungo".

! Attacco l'albero di rango minore, come figlio di quello di rango maggiore.



Con l'unione per rango:

① MAKESET (x): $P(x) = x$, $Rank(x) = 0$ → Ad ogni nodo associo un rango.

(2) FINDSET(X) :

(3) UNION(x, y) : 2 alberi in input

Link(FindSet(x), FindSet(y))

Link(x, y):

If rank(x) > rank(y) :

P(y) = x ;

else

P(x) = y ;

If rank(x) == rank(y)

rank(y) ++ → quando i ranghi sono uguali

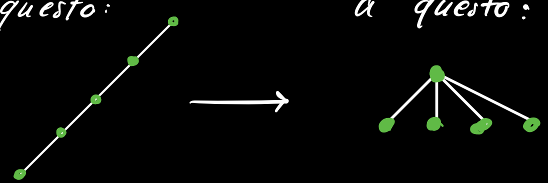
Miglioramento della findset(): (ho un algoritmo ricorsivo)

if x ≠ p(x)

then P(x) = Findset(x)

Return P(x) ;

Passo da questo: a questo:



quando voglio il parenti

la prima findset() → Θ(n), successivamente taglio il tempo della findset.

- TEMPISTICHE:

Unione per rango → $O(n \log(n))$

\+

Compressione dei cammini → $O(m \cdot \partial(m, n))$ $\partial \leq 4$

non l'ha spiegata molto bene.

Algoritmo Kruskal (minimun spanning tree):

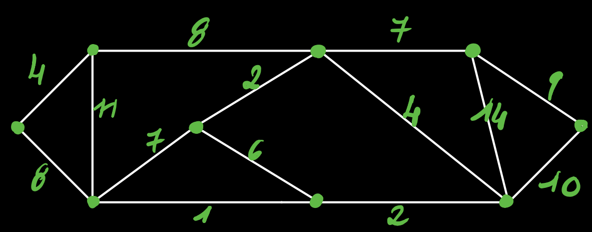
- Algoritmo Greedy
- Sfrutta gli insiemi disgiunti.

Dato $G = (V, E)$, PESATO (sui lati), CONNESSO e NON ORIENTATO.

$\Rightarrow$ Ottengo un: $T \subseteq E$ t.c. $G' = (V, T)$ sia ACICLICO, CONNESSO, DI PESO MINIMO.

↳ percorso.

ESEMPIO:



→ Voglio il suo MST

→ Ossia generico greedy

Generic MST (G, w):

$A = \emptyset$

While A non é MST

Trova_Arco (u, v) da aggiungere

$A = A \cup \{(u, v)\}$

Return (A)

- **Arco sicuro** → × UN ARCO, CHE SE AGGIUNTO AD A, MI GARANTISCE CHE A

(sicuro)

SIA ANCORA UN MST CON LE INFORMAZIONI TROVATE FINO ALLA SCOPERTA DI x

KRUSKAL:

- A è una foresta. → vertici presi da G ⇒ V.
- Ad ogni passo: Aggiunge un arco sicuro z = (u, v).
- z: è un arco di peso minimo che collega due componenti distinte.

↳ è sempre sicuro (dimostrazione sul libro).

MST_KRUSKAL (G, w)

$A = \emptyset$

$\forall v \in V$, makeset(v) → setting iniziale

$E' =$ sort (w(E))

$\forall (u,v) \in E'$

if Findset(u) $\neq$ findset(v)

$A = A \cup \{(u,v)\}$

Union (u,v)

Return (A).

Tempo : $O(|E| \log |E|)$