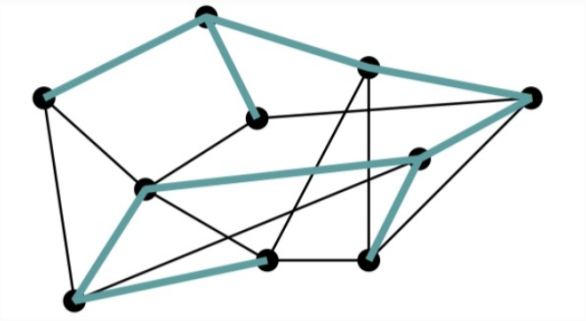
Sia dato un grafo connesso e non orientato

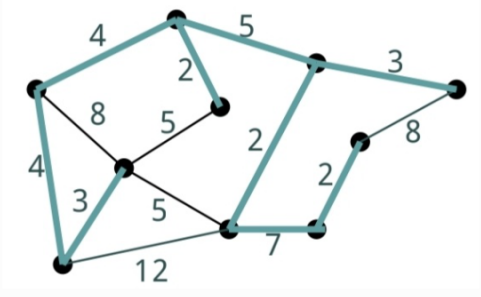
ALBERO RICOPRENTE = sottografo
- contenente tutti i nodi
- aciclico
- connesso



Dato un grafo pesato $G(V,E)$ non orientato,

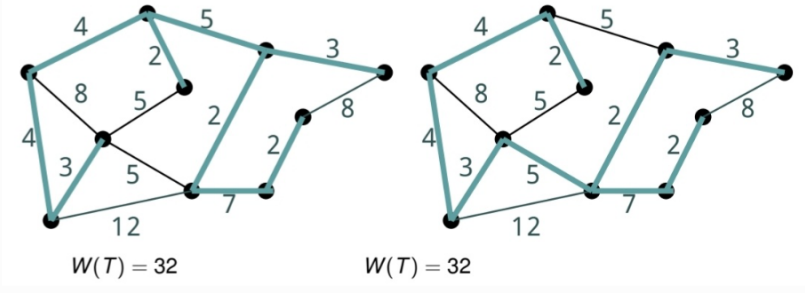
l' ALBERO MINIMO RICOPRENTE è l'albero T con peso minimo

esempio



T t.c. $W(T)$ è minimo : $W(T) = 32$

N.B. non è unico



ALGORITMO GENERICO :

```
MINIMO_ALBERO_RICOPRENTE(G)
A ← ∅
while A non è un albero ricoprente do
    trova un arco (u, v) che sia "sicuro" per A
    A ← A ∪ (u, v)
```

?

↓

Al termine, $T = (V, A)$ è un albero minimo ricoprente

- Come si trova un arco sicuro ?

Definizione Taglio di $G(V,E)$: partizione di V in due insiemi X e $V \setminus X$

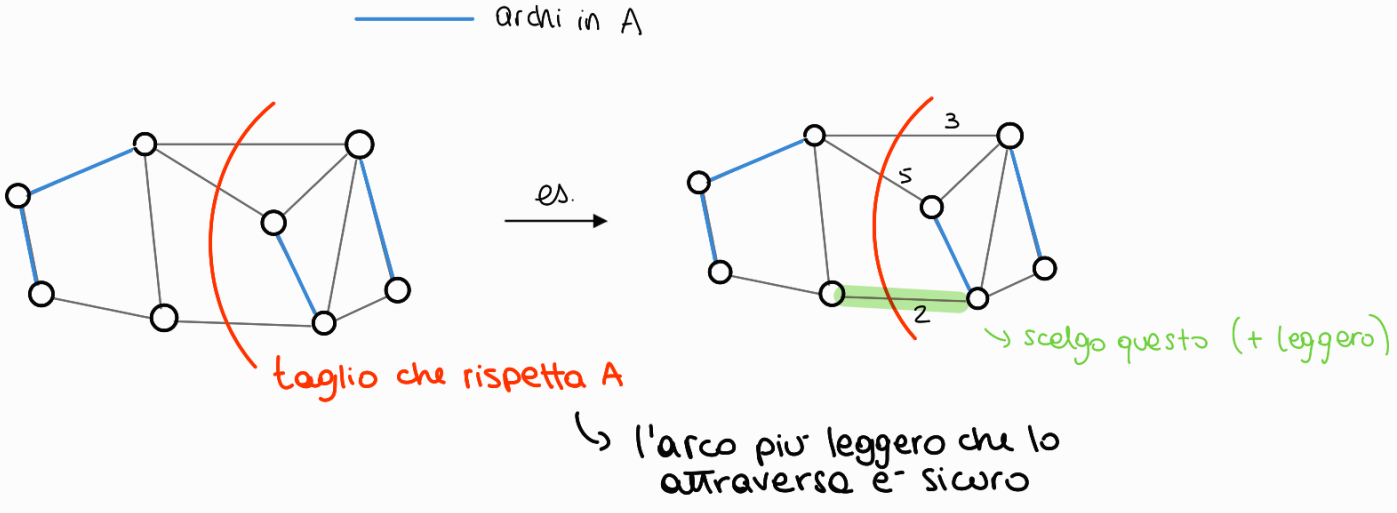
- l'arco (u,v) attraversa il taglio $(X, V \setminus X)$ se $u \in X$ e $v \in V \setminus X$
- un taglio rispetta un insieme di archi A se nessun arco di A attraversa il taglio
- un arco che attraversa un taglio è leggero se è un arco di peso minimo tra quelli che attraversano il taglio

CRITERIO PER ARCHI SICURI

- Teorema

Sia :
- $G(V, E)$ connesso, non orientato e pesato
- $A \subseteq E$ contenuto in un albero minimo ricoprente
- $(X, V-X)$ un taglio che rispetta A
- (u,v) un arco leggero che attraversa $(X, V-X)$

$\Rightarrow$ (u,v) è sicuro per A



↳ l'arco più leggero che lo attraversa è sicuro

- **Corollario**

Sia :
- $G(V, E)$ connesso, non orientato e pesato
- $A \subseteq E$ contenuto in un albero minimo ricoprente
- C componente connessa (un albero) nella foresta $G = (V, A)$
- (u, v) un arco leggero che connette C a una qualche altra componente connessa di $G(A)$

$\Rightarrow$ (u, v) è sicuro per A

Da qui si basano due algoritmi per la scelta dell'arco sicuro : - Kruskal
- Prim

Algoritmo di Kruskal

```
MST_Kruskal(G)
  A ← ∅
  ordina gli archi in ordine non decrescente di peso
  for ∀(u, v) nell'ordine do
    if (u, v) è "sicuro" per A then
      A ← A ∪ (u, v)
```

$O(|E| \log |E|)$ ←

Non serve un nodo da cui partire

Obiettivo: costruzione di un albero (grafo connesso aciclico)

→ Bisogna memorizzare i vertici delle componenti connesse del sottoinsieme dell'albero di copertura in costruzione

→ Serve una struttura dati per gli insiemi disgiunti

collezione S di insiemi disgiunti

la scelta di un arco crea un ciclo se unisco due insiemi non disgiunti

affinché gli archi scelti non creino cicli : Unione di insiemi DISGIUNTI

all'inizio, ogni insieme contiene un singolo nodo

l'algoritmo diventa

```
MST_Kruskal(G)
  A ← ∅
  for ∀v ∈ V do
    Make_set(v)        crea insieme con unico elemento v
  ordina gli archi in ordine non decrescente di peso
  for ∀(u, v) ∈ E nell'ordine do
    if Find(u)≠Find(v) then    trova il rappresentante dell'insieme contenente v
      A ← A ∪ (u, v)
      Union(u, v)      unisce gli insiemi che contengono u e v
```

$O(|E| \log |V|)$

Algoritmo di Prim

```
MST_Prim(G, s)
  A ← ∅
  Q ← V − {s}
  while Q ≠ ∅ do
    scegli un arco (u, v) "sicuro" per A
    A ← A ∪ (u, v)
    Q ← Q − {v}
```

s: nodo di partenza

Strategia: mantenere un sottografo connesso

Q: nodi che devo ancora ricoprire

togli v da Q perché è stato coperto

Come scelgo il nodo sicuro? Memorizzo ad ogni iterazione quali sono gli archi usati per raggiungere un certo nodo

```
MST_Prim(G, s)
  Q ← V
  for ∀v ∈ V do v.d ← ∞
  s.d ← 0
  s.π ← nil
  while Q ≠ ∅ do
    u ← nodo con d minimo in Q (tolto da Q)
    for ∀v ∈ adj[u] do
      if v ∈ Q e W(u, v) < v.d then
        v.d ← W(u, v)
        v.π ← u
```

peso iniziale degli archi

il nodo iniziale è già coperto

Q può essere implementata come una coda di priorità con heap min data da d

$O(|E| \log |V|)$