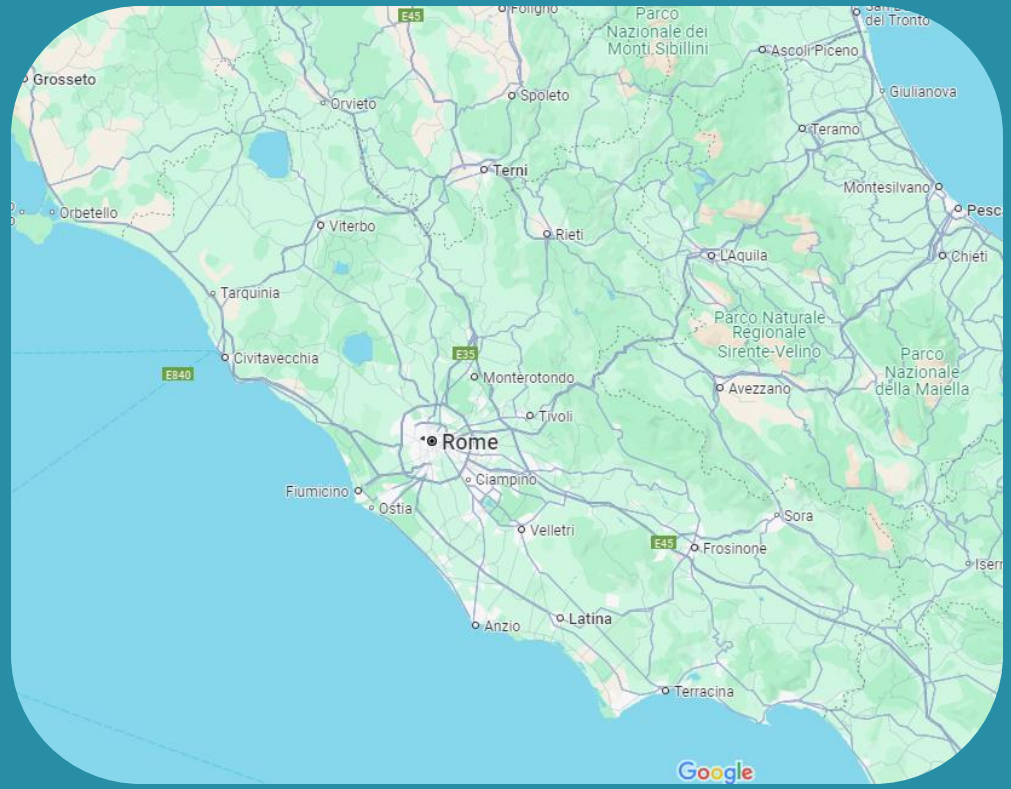
COME FUNZIONANO I NAVIGATORI

Tutti abbiamo almeno una volta utilizzato un navigatore come GoogleMaps o AppleMaps.

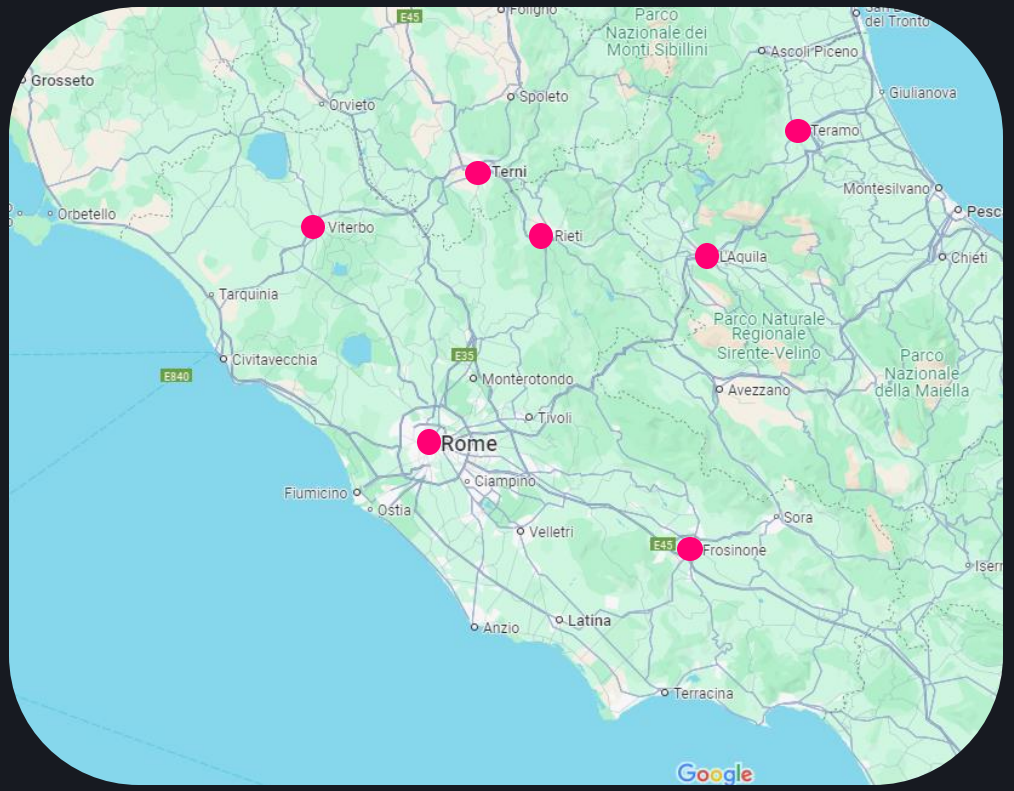
Ma come funzionano veramente?

Come fanno a portarci da un punto A ad un punto B percorrendo meno strada possibile?

Oggi proveremo a capire il funzionamento di un semplice navigatore con l'aiuto di una struttura dati chiamata grafo



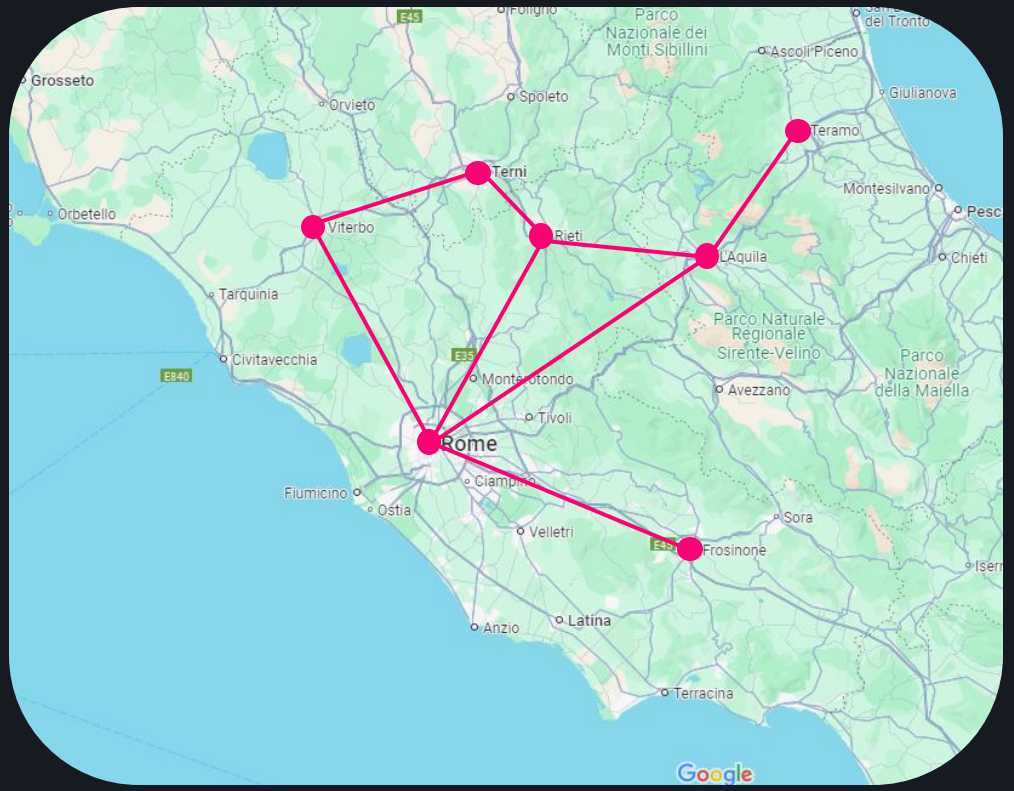
I NODI



Assegniamo a ogni città un nodo, questi saranno le località nelle quali il nostro navigatore ci potrà portare o dalle quali partiremo

Un nodo è l'unità fondamentale di cui i grafi sono costituiti: un grafo consiste in un insieme di vertici e di archi

GLI ARCHI



Ora uniamo i diversi nodi tramite degli archi.
In questo caso gli archi rappresentano le strade che possiamo percorrere per spostarci da una località all'altra

Un arco in un grafo è una connessione tra due nodi (o vertici).

Di solito si rappresenta con una linea che collega i due nodi

TIPOLOGIE DI ARCHI

Archi orientati

Possiamo andare solo dal punto A al punto B e non viceversa

Esempio: Una strada a senso unico

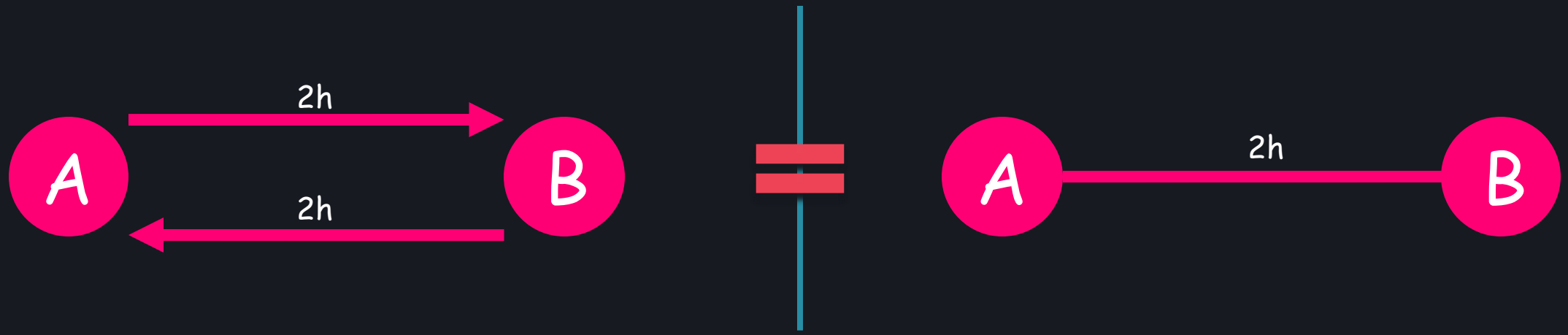
Archi Non orientati

Possiamo percorrere l'arco in entrambe le direzioni, da A a B e viceversa

Esempio: Una strada a doppio senso

TIPOLOGIE DI ARCHI

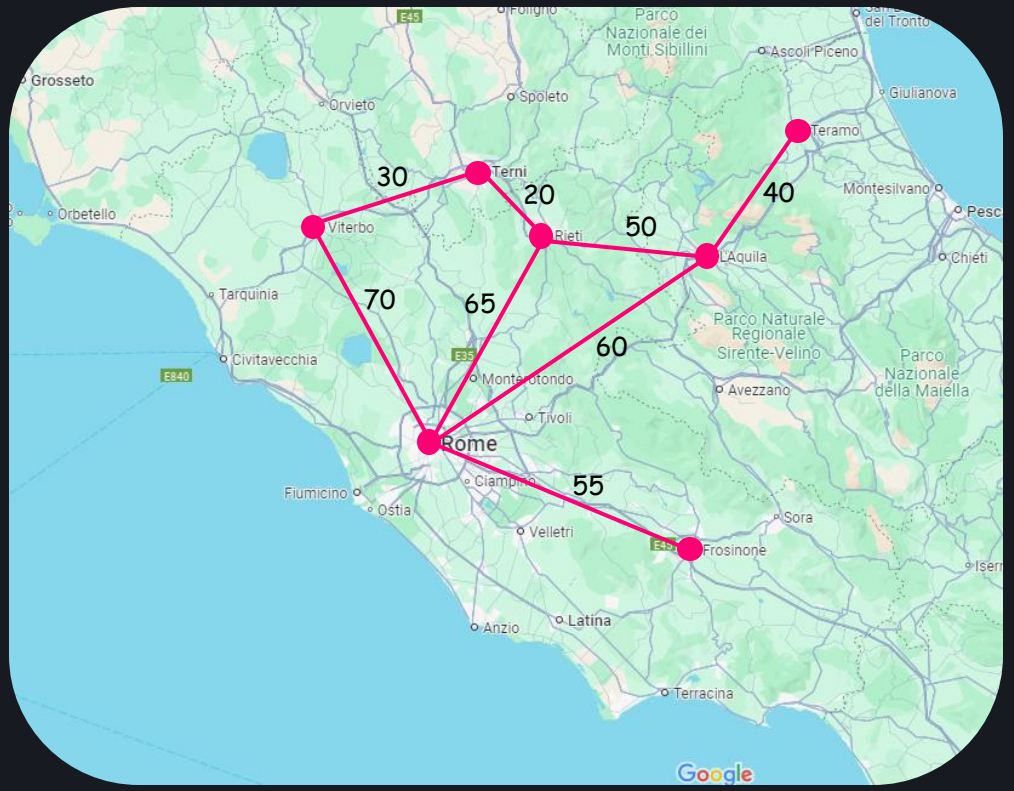
E se metto 2 archi orientati in direzioni opposte?



I due grafi sono uguali solo se le coppie di archi orientati in direzioni opposte hanno il peso dello stesso valore

Pensare al peso come la lunghezza della strada o il tempo di percorrenza

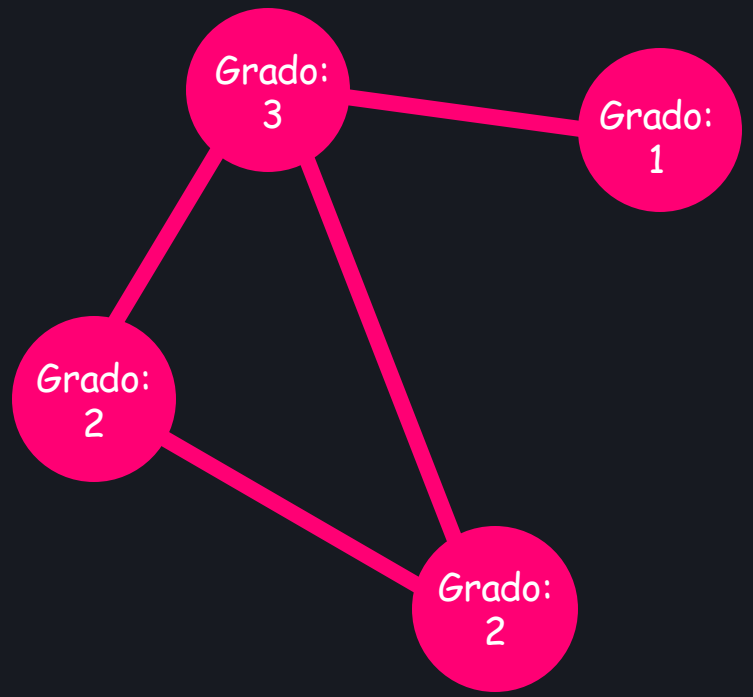
I PESI



Per andare da una città all'altra dovremmo percorrere n km.
In questo caso rappresentiamo i pesi degli archi come i km da percorrere

Un grafo pesato associa un peso ad ogni suo arco.
I pesi sono espressi generalmente tramite numeri reali, ma possono essere ristretti all'insieme dei razionali o degli interi.

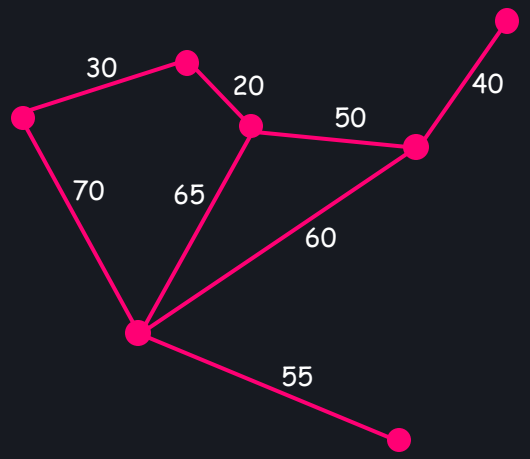
GRADO



Notiamo, banalmente, che in questo esempio ogni nodo è collegato a un numero specifico di archi, questo si chiama grado

Numero di archi *k* collegati a quel nodo

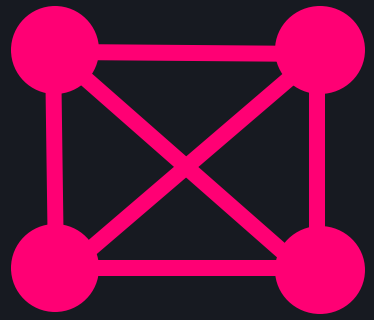
IL GRAFO



Benissimo,
abbiamo creato il nostro grafo pesato non orientato

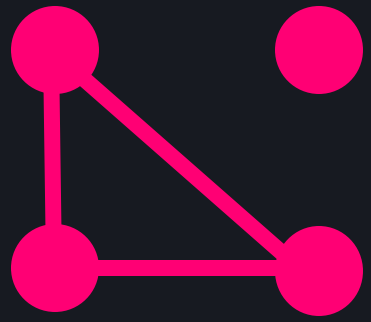
TIPOLOGIE DI GRAFI

Grafo completo



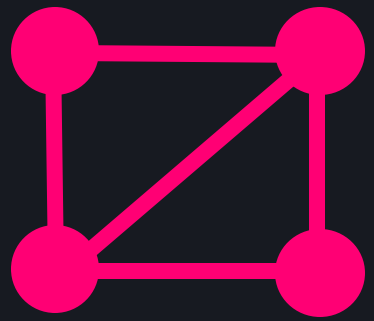
Ogni nodo è collegato a tutti gli altri

Grafo disconnesso



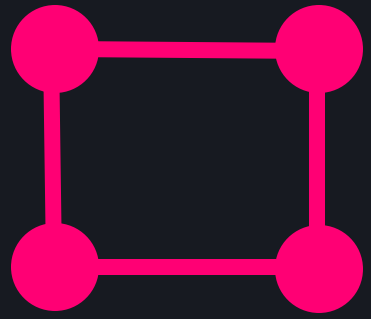
Esistono nodi che non sono collegati tra loro da alcun arco.

Grafo planare



Può essere disegnato su un piano senza che i suoi archi si intersechino

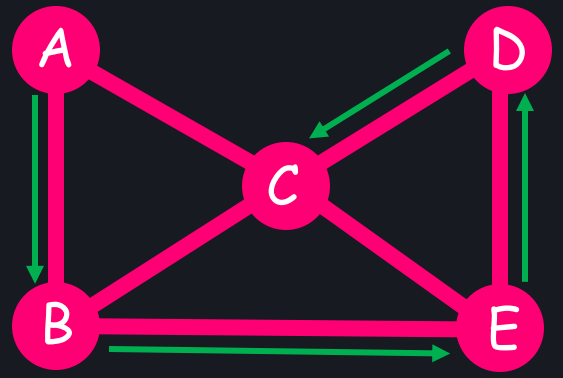
Grafo regolare



Tutti i nodi hanno lo stesso numero di archi incidenti, ossia lo stesso grado

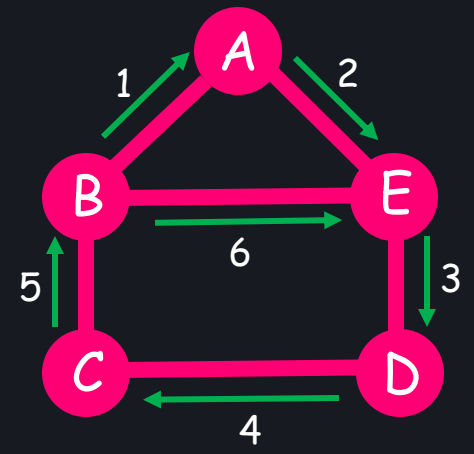
I CAMMINI

Cammino Hamiltoniano



Un cammino hamiltoniano in un grafo è un percorso che visita ogni ***nodo*** esattamente ***una volta***, senza necessariamente attraversare tutti gli archi.

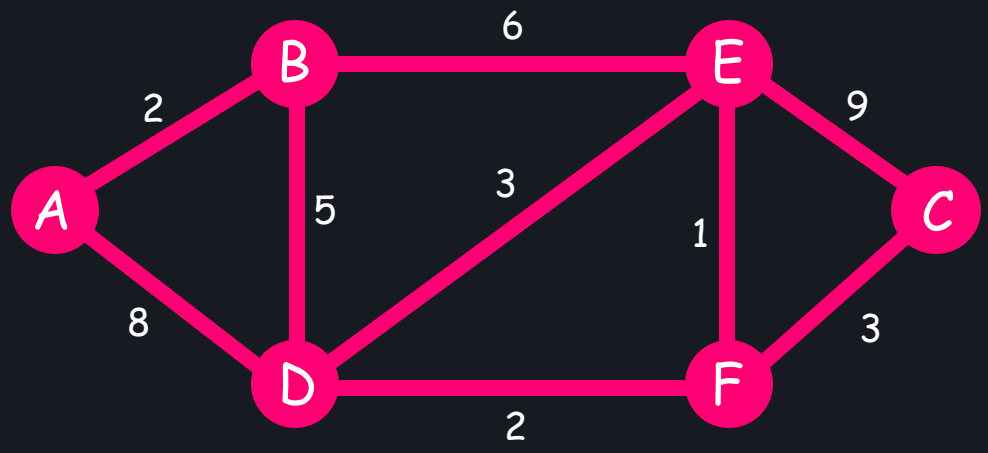
Cammino Euleriano



Un cammino euleriano in un grafo è un percorso che attraversa ogni ***arco*** esattamente ***una volta***

ESEMPIO PRATICO

Source: link video



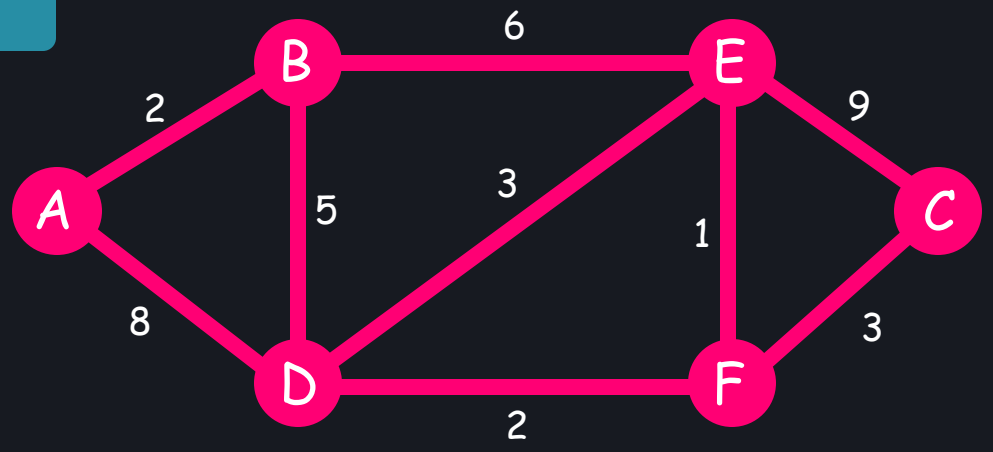
Facciamo finta che il grafo connette più località e i pesi sono i tempi di percorrenza

Usiamo l'algoritmo di Dijkstra per calcolare il percorso più breve

Il teorema di Dijkstra descrive un algoritmo che trova il percorso più breve da un nodo sorgente a tutti gli altri nodi in un grafo pesato. Funziona selezionando progressivamente il nodo con la distanza minima non ancora visitato.

ESEMPIO PRATICO

Source: link video



Nodo	Ore	Nodo precedente
A	0	-
B	∞	-
C	∞	-
D	∞	-
E	∞	-
F	∞	-

Per eseguire l'algoritmo ci occorre:

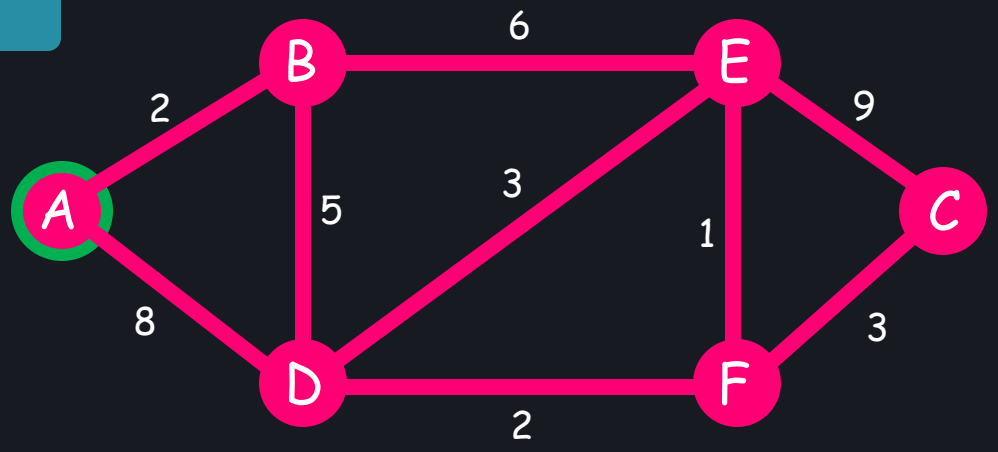
- Una tabella
- Tenere traccia dei nodi visitati

Algoritmo:

1. Settiamo a 0 il nodo da cui partiamo, tutti gli altri a infinito
2. Scegliamo un nodo non visitato che ha il tempo minore nella tabella
3. Calcola le ore con i nodi adiacenti
4. Contrassegna il nodo scelto come visitato
5. Ripeti i procedimenti 2-3-4 fino a che non ci son più nodi raggiungibili

ESEMPIO PRATICO(1)

Source: link video



Nodo	Ore	Nodo precedente
A	0	-
B	2	A
C	∞	-
D	8	A
E	∞	-
F	∞	-

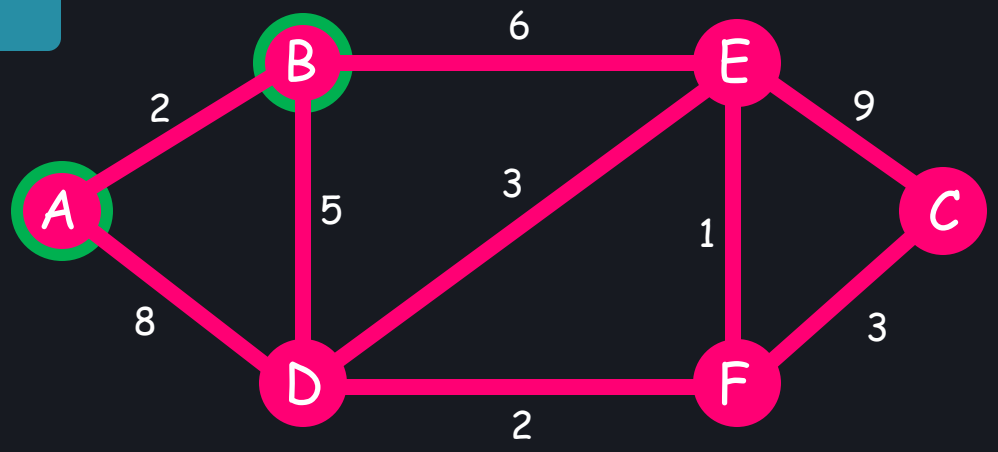
Il nodo non visitato con tempo minore è banalmente A, partiamo da lui e visitiamo i suoi nodi adiacenti

Algoritmo:

1. Settiamo a 0 il nodo da cui partiamo, tutti gli altri a infinito
2. Scegliamo un nodo non visitato che ha il tempo minore nella tabella
3. Calcola le ore con i nodi adiacenti
4. Contrassegna il nodo scelto come visitato
5. Ripeti i procedimenti 2-3-4 fino a che non ci son più nodi raggiungibili

ESEMPIO PRATICO(2)

Source: link video



Nodo	Ore	Nodo precedente
A	0	-
B	2	A
C	∞	-
D	5 + 2 < 8	B
E	6 + 2	B
F	∞	-

Fra B e D il nodo con tempo minore è B, partiamo da lui e visitiamo i nodi adiacenti non ancora visitati

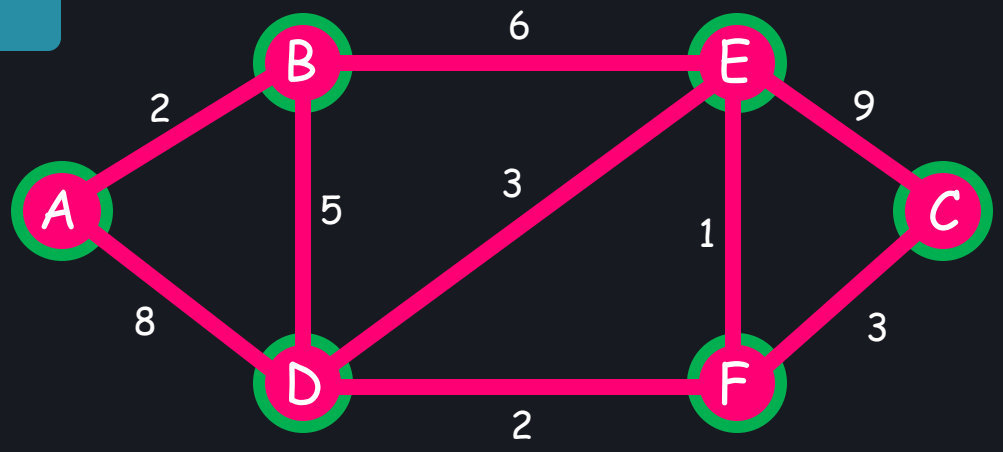
Algoritmo:

1. Settiamo a 0 il nodo da cui partiamo, tutti gli altri a infinito
2. Scegliamo un nodo non visitato che ha il tempo minore nella tabella
3. Calcola le ore con i nodi adiacenti
4. Contrassegna il nodo scelto come visitato
5. Ripeti i procedimenti 2-3-4 fino a che non ci son più nodi raggiungibili

Per inserire nuova distanza:
nuovo < vecchio

ESEMPIO PRATICO(3)

Source: link video



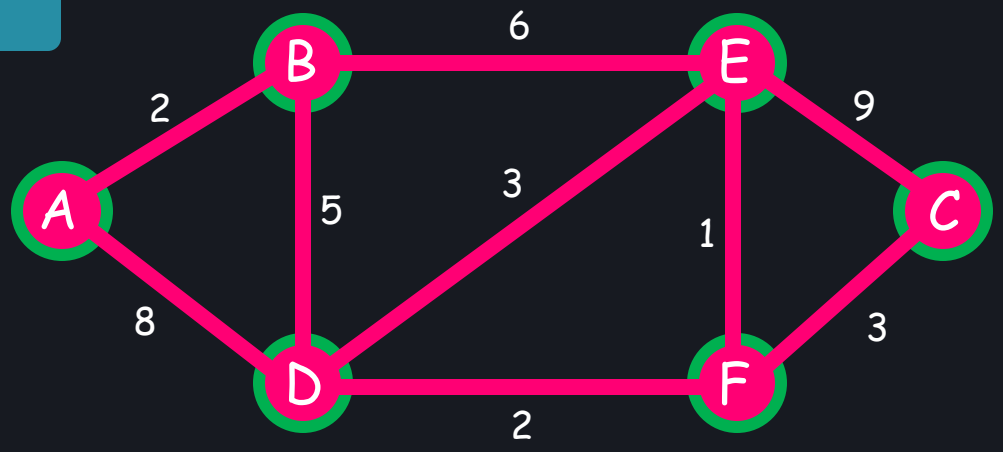
Nodo	Ore	Nodo precedente
A	0	-
B	2	A
C	12	F
D	7	B
E	8	B
F	9	D

Ora che più o meno abbiamo capito l'algoritmo saltiamo al risultato finale, cioè quando non abbiamo più nodi raggiungibili

Ora da questa tabella riusciremo a trovare il percorso minimo utilizzando la colonna del nodo adiacente, procedendo a ritroso

ESEMPIO PRATICO(4)

Source: link video



Nodo	Ore	Nodo precedente
A	0	-
B	2	A
C	12	F
D	7	B
E	8	B
F	9	D

Ora che più o meno abbiamo capito l'algoritmo saltiamo al risultato finale, cioè quando non abbiamo più nodi raggiungibili

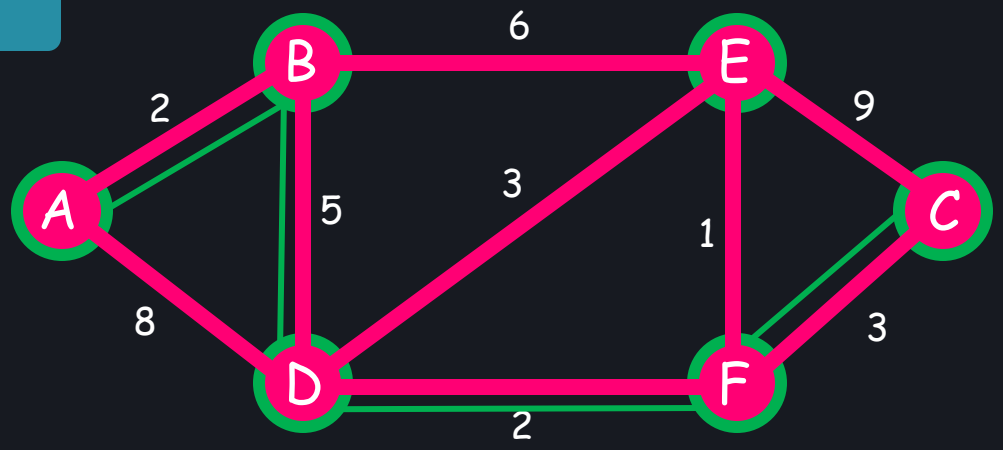
Ora da questa tabella riusciremo a trovare il percorso minimo utilizzando la colonna del nodo adiacente, procedendo a ritroso

Vogliamo trovare il percorso più breve per arrivare al punto C da A

Soluzione:
A -> B -> D -> F -> C

ESEMPIO PRATICO(5)

Source: link video



Nodo	Ore	Nodo precedente
A	0	-
B	2	A
C	12	F
D	7	B
E	8	B
F	9	D

Vogliamo trovare il percorso più breve per arrivare al punto C da A

Soluzione:
A -> B -> D -> F -> C

Perché?
Partendo da C abbiamo i seguenti nodi precedenti:

- Precedente a C -> F
- Precedente a F -> D
- Precedente a D -> B
- Precedente a B -> A

Totale ore di viaggio minime: 12h