

# Distance Vector Routing

Il seguente file spiega il funzionamento dello script **code.py**

## -Logica dietro lo script

Come è noto, il distance vector routing si basa sul principio che una rete può essere trattata come un **grafo**, e che la distanza minima tra nodi del grafo è trovabile con l'applicazione di un algoritmo shortest-distance (nel nostro caso Bellman-Ford).

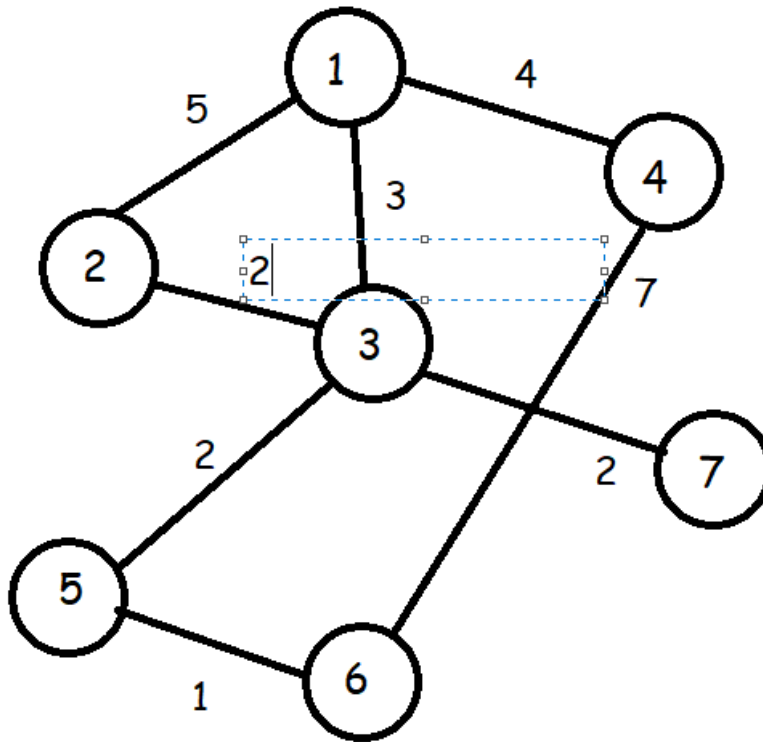
Bisogna quindi implementare una soluzione che:

- Tratti i router di rete come nodi di un grafo
- Tratti i collegamenti ai router come gli archi di un grafo
- Esegua l'algoritmo shortest distance
- Costruisca le routing table step by step

## -Implementazione

Lo script è diviso in 3 parti:

- L'implementazione della classe **routing\_node**: questa classe implementa il singolo nodo di rete. Questa classe dispone dei metodi per:
  - 1)Inizializzarsi
  - 2)Aggiungere alla routing table un collegamento diretto
  - 3)Aggiornare la propria tabella di routing (elemento fondamentale dell'algoritmo DV)
  - 4)Stampare il proprio contenuto
- L'implementazione della classe **network\_manager**: questa classe è invece responsabile di impostare i collegamenti tra i nodi ed eseguire il Distance Vector.
- Un piccolo metodo main che crea un'istanza di network manager e crea la rete nell'immagine



La parte più importante del codice è decisamente il metodo **run\_DV** di network manager, che esegue effettivamente l'algoritmo.

Lo script esegue N round di aggiornamenti, dove N è il numero di nodi di Rete.

Ad ogni round ogni nodo di rete aggiorna la sua routing table.

### **-Utilizzo dello script**

Lo script python mostra il Distance Vector sul grafo di esempio.

Eseguire lo script è sufficiente a visualizzare le routing table di tutti i nodi.

Eventualmente modificare il main permette di testare su reti diverse l'algoritmo.