

Cammino Minimo

Problema: Dato un **grafo orientato** $G = (N, A)$ ed **etichettato** negli archi **con valori interi positivi**, trovare il cammino minimo.

Questo è un problema di minimizzazione, che rientra nei **problemi di ottimizzazione**.

Risoluzione con Algoritmo di Dijkstra

Una delle tecniche per trovare il cammino minimo in un grafo consiste nell'usare l'**algoritmo di Dijkstra**.

Quest'algoritmo può essere usato con grafi orientati e pesati (con pesi positivi).

L'algoritmo usa:

- una *coda di priorità* per memorizzare coppie $\langle \text{indice}, \text{distanza} \rangle$;
- un *vettore delle distanze*, nel quale i nodi corrispondono agli indici dell'array.

Prima di iniziare la ricerca l'algoritmo inizializza il vettore delle distanze a $+\infty$ per tutti i nodi tranne che per il nodo sorgente (che verrà inizializzato a 0), che sarà il nodo dal quale verranno calcolate le distanze; la scelta di questo nodo è a discrezione dell'utente.

Si inserisce inoltre la coppia $\langle \text{indice nodo sorgente}, 0 \rangle$ nella coda, che sarà il primo elemento ad essere usato.

L'utilizzo della distanza $+\infty$ assume che tutti i nodi siano irraggiungibili inizialmente; ciò significa che, se alla fine del processo ci sono nodi ancora con quel valore, questi sono irraggiungibili.

L'algoritmo inizia la ricerca e segue questi steps:

1. Legge la coppia $\langle \text{indice}, \text{distanza} \rangle$ dalla coda (essendo una coda di priorità sceglierà quella con chiave/distanza minore.
Se il valore *distanza* della coda è maggiore a quello contenuto nell'array la coppia viene rimossa e si sceglie la prossima, altrimenti si prosegue;
2. Sceglie un vicino del nodo e calcola la distanza del percorso dal nodo letto dalla coda al vicino e se questo è inferiore di quello nell'array lo aggiorna; inserisce inoltre la coppia $\langle \text{indice vicino}, \text{distanza} \rangle$ nella coda.
Prosegue fin quando tutti i vicini sono stati visitati.
Terminato questo step rimuove la coppia letta dalla coda.
3. Ripete gli steps 1 e 2 fin quando la coda risulta vuota.

L'algoritmo restituirà in **output** il vettore contente i costi/distanze per tutti i cammini a partire dal nodo sorgente. In base all'implementazione sarà possibile anche mostrare i cammini relativi alle distanze nel vettore.