Argomenti Orale Algoritmi 1

# Algoritmo di Kosarasu

L’algoritmo di Kosarasu è un algoritmo che permette di trovare le componenti fortemente connesse in un grafo orientato, esso le sa attraverso tre passaggi:

1. Calcolo dell’ordine di fine visita (ordine in cui terminano i nodi, equivalente all’ordine topologico al contrario se il grafo è aciclico) con la visita DFS (non si può ottenere con la visita BFS);
2. Calcolo del grafo trasposto, ovvero un grafo con lo stesso numero di nodi del precedente ma con tutti gli archi invertiti;
3. Visita del grafo trasposto utilizzando l’ordine di fine visita (la seconda visita è indifferente perché si basa su quest’ultimo).

## Visita BFS

La visita BFS è un tipo di visita di un grafo in cui vengono scoperti e visitati tutti i vicini del nodo dato per poi proseguire fino a quando non li ho scoperti tutti. Per fare ciò, la visita si serve di una coda in cui memorizzare ed estrarre i nodi da visitare. Una proprietà di questa visita è che i nodi inseriti nella coda corrispondo a quelli del livello successivo dell’albero di visita rispetto a quello che si sta visitando.

## Visita DFS

La visita DFS è un tipo di visita in cui i nodi vengono scoperti a partire dal “fondo” e man mano risalendo il percorso fatto dal grafo. L’algoritmo, alla scoperta di un nuovo nodo, esegue una chiamata ricorsiva su esso e mettendo in pausa la visita del precedente (la visita però non è ancora terminata). È possibile sfruttare questa peculiarità utilizzandola quando si è in presenza di cicli.

## Rilevazione di cicli nei grafi

La rilevazione di cicli in grafi orientati e non è possibile attraverso gli alberi di visita ricavabili dalle visite, da essi infatti si può constatare che:

* L’albero di visita di un grafo orientato è formato da 3 tipi di archi tra quelli non considerati dall’albero: gli archi in avanti/all’indietro permettono di salire/scendere di livello mentre quelli trasversali connettono due nodi fratelli.
* Dal punto precedente, si può dire che un grafo orientato ha un ciclo se è presente un arco all’indietro che connette due nodi.
* In un grafo non orientato, l’albero di visita non considera solo gli archi trasversali o quelli in avanti/all’indietro a seconda dell’algoritmo di vista utilizzato (rispettivamente BFS per il primo caso e DFS per il secondo), si può quindi dire che un grafo ha un ciclo se l’albero BFS non considera un arco trasversale oppure se l’albero DFS non ne considera uno in avanti/all’indietro.
* La peculiarità della visita DFS permette la rilevazione di cicli in ogni tipo di grafo.

La ricostruzione d’un ciclo avviene tramite l’array in padri, in cui si continua a inserire nodi finché non arrivo alla radice oppure arrivo al padre del nodo di partenza.

## Ordine Topologico

L’ordine Topologico è una rappresentazione del grafo orientato in cui i nodi vengono disposti in odo da avere tutti gli archi che partono da un nodo di sinistra e arrivano a un di destra. Per permettere ciò, il grafo deve essere aciclico, ovvero non deve presentare cicli al suo interno (DAG).

## Componenti Fortemente Connesse (SCC)

Una Componente Fortemente Connessa di un grafo orientato è una componente in ogni per ogni coppia di nodi appartenente a essa, esiste un cammino orientato. Sfruttando queste componenti è possibile, dato un grafo orientato, ricavare il grafo delle SCC, ovvero una rappresentazione del grafo originale in cui si utilizzano le SCC come singoli nodi. Il grafo delle SCC è sempre DAG e ha la seguente proprietà: se esiste un arco che parte da una SCC A e arriva in B, l’ultimo nodo a terminare sarà in A, questo succede perché.

# Problema del commesso viaggiatore

Il problema del commesso viaggiatore (TSP) è un problema difficile in cui, dato un grafo non orientato, completo e con pesi positivi, bisogna percorrere tutti i nodi minimizzando le distanze. Nel suo complesso TSP non ammette approssimazioni per nessuna R, esiste però un caso in cui si può ottenere una soluzione vicina a quella ottimale: il TSP metrico, ovvero il TSP in cui è presente la disuguaglianza triangolare applicata agli archi (il lato più lungo di un triangolo deve essere minore o uguale della somma degli altri 2).

Supponiamo che S\* sia la soluzione ottimale del TSP metrico. Dato che S\* è un cammino chiuso, se togliamo un arco qualsiasi da esso, si ottiene un albero ricoprente T. Il costo di T è sempre minore o uguale a quello di S\*, dato che, togliendo un arco, tutti gli archi sono positivi e possono toglierne uno di peso 0 nel caso limite.

( c(S\*) >= c(T) )

Un primo tentativo d’approssimazione si può fare calcolando MST del grafo (A): in questo modo però il commesso percorrerebbe tutti gli archi 2 volte, dato che A non è un cammino chiuso. Dato che A è MST, si può dire che c(T) >= c(A) perché A è il più piccolo tra tutti gli alberi ricoprenti.

Prendiamo ora il cammino più lungo che il commesso può percorrere (W) e, confrontandolo col cammino di A, si ha che c(W)= 2\*c(A) per il motivo spiegato in precedenza. Dato che il grafo è completo, è possibile “tagliare” e, se si può fare, conviene farlo? Sì e in questo modo la lunghezza del tour del commesso diminuisce. Con il nuovo percorso tagliato (W’) si ha che c(W’) <= c(W) per disuguaglianza triangolare.

Dato che prendere le scorciatoie conviene, prendiamole sempre! Per farlo si utilizza l’ordine dei nodi scoperti della visita DFS, in questo modo si ottiene un tour S in cui c(S) <= c(W)<=2\*c(A) dato che prendo tutte le scorciatoie per diminuire la distanza. Dato che, come detto prima, c(S\*)>=c(T)>=c(A), si può dire che c(S)<=c(W)=2\*c(A)<=2\*c(S\*). In conclusione, dato che il tour di S non può essere peggiore del doppio di quello di S\*, si può dire che c(S)<=2\*c(S\*) in cui 2 è il cosiddetto fattore d’approssimazione. Tuttavia non riesco a dimostrare se esiste un percorso migliore calcolabile col medesimo modo ma, dato che MST non ha a che fare con i cammini minimi, non riuscirei a dimostrare che la soluzione trovata sia ottimale.