Esercitazione 2: More Depth-First Search

Giacomo Paesani

March 14, 2024

Esercizio 1. In un grafo non diretto e connesso $G = (V, E)$ un vertice v si dice di *articolazione* (*cutvertex* in inglese) se $G - v$, il grafo ottenuto da G rimuovendo v e tutti gli archi ad esso incidenti, non è connesso. Modificare l'algoritmo della ricerca in profondità in maniera da poter ottenere tutti e soli i vertici di articolazione di un grafo connesso; è possibile fare questa modifica in modo che il controllo avvenga in $\Theta(|V| + |E|)$?

Come si può ulteriormente modificare l'algoritmo per ottenere tutti i vertici di articolazione di un grafo non necessariamente connesso (dove in questo caso il criterio di un vertice di articolazione è quello di disconnettere la componente connessa che lo contiene)?

Esercizio 2. In un grafo non diretto G un cammino *Hamiltoniano* è un cammino P di G che passa per ogni vertice di G esattamente una volta. Provare o confutare la seguente affermazione: G contiene un cammino Hamiltoniano se e solo se può essere prodotto un albero di ricerca di G che è un cammino. Domanda bonus: che succede se invece del cammino Hamiltoniano, si ha un ciclo Hamiltoniano in G?

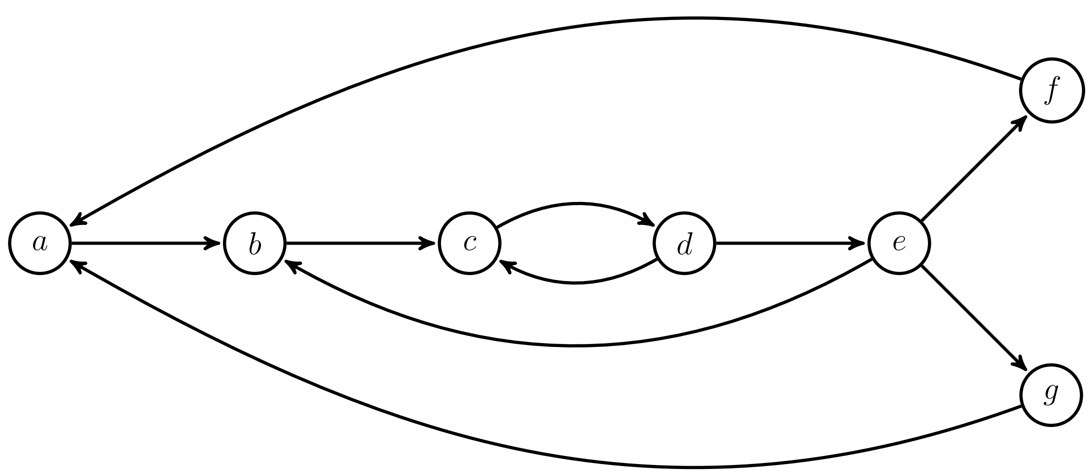
Esercizio 3 (22.3-1,[1])**.** Durante una visita in profondità, ad ogni vertice u di un grafo viene associato un colore che varia nel tempo t dell'algoritmo. Se $t < t[u]$ e cioè il vertice u deve ancora essere stato visitato per la prima volta allora u è colorato di BIANCO. Se $t[u] \leq t \leq T[u]$ e cioè il vertice u è ancora in visita allora u è colorato di GRIGIO. Infine, se $t > T[u]$ e cioè il vertice u è stato già completamente visitato allora u è colorato di NERO. Fare una tabella 3x3 con righe e colonne contrassegnate da BIANCO, GRIGIO e NERO. In ogni cella (i, j), indicare se, in un alcun punto di una ricerca in profondità di un grafo diretto, è possibile avere un arco da un vertice di colore i ad un vertice di colore j. Per ogni possibile arco, indicare se esso può essere:

- arco dell'albero,
- arco all'indietro,
- arco in avanti o
- arco di attraversamento.

Fare una seconda tabella per una ricerca in profondità di un grafo non diretto.

	BIANCO	GRIGIO	NERO
BIANCO			
GRIGIO			
NERO			

Esercizio 4 (I. Salvo)**.** Sia G il grafo raffigurato in figura. Determinare il minimo numero di archi che devono essere eliminati da G affinchè G ammetta ordinamenti topologici. Una volta rimosso questo insime minimo di archi, determinare tutti gli ordinamenti topologici di G.



Esercizio 5. Fornire un algoritmo in pseudo-codice che dato un grafo diretto e aciclico $G = (V, E)$, restituisce un ordinamento topologico di G. E' possibile implementarlo in modo che il tempo di esecuzione sia $\Theta(|V| + |E|)$? Come dovrebbe essere modificato tale algoritmo per restituire l'elenco di tutti gli ordinamenti topologici di G? E' possibile fare questa ulteriore modifica mantenendo lo stesso tempo di esecuzione?

Esercizio 6 (I. Salvo)**.** Descrivere in pseudo-codice un algoritmo che, dato un grafo non diretto G, descrivere un algoritmo che ne orienta gli archi in modo da creare un grafo G' diretto e aciclico. Questo algoritmo deve avere tempo di esecuzione $\Theta(n + m)$.

References

[1] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms. 2022.

[2] Arthur B. Kahn. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562, 1962.