Esercitazione 3: Strongly Connected Components

Giacomo Paesani

April 10, 2024

Esercizio 1 (22.2-9, [1])**.** Fornire una algoritmo in pseudo-codice che, dato un grafo non diretto e connesso $G = (V, E)$, trova una passeggiata in G che attraversa tutti gli archi una e una sola volta in ognuna della due direzioni in tempo $\mathcal{O}(|V| + |E|)$.

Esercizio 2 (22.4-2, [1])**.** Fornire un algoritmo in pseudo-codice che dato un grafo diretto e aciclico $G = (V, E)$ e due vertici s e t, restituisce il numero di tutti i cammini da s a t in G.

Esercizio 3. Sia un grafo diretto $G = (V, E)$ ed s e t due vertici di G. G si dice *s-t-connesso* se ogni vertice di G è in almeno un cammino da s a t. Fornire un algoritmo in pseudo-codice che dato un grafo diretto e aciclico $G = (V, E)$ ed s e t due vertici di G, restituisce un sottografo G' di G massimale s-t-connesso.

Esercizio 4 (22.5-1, [1])**.** Come cambia il numero di componenti fortemente connesse di un grafo diretto se viene aggiunto un nuovo arco?

Esercizio 5 (I. Salvo)**.** Sia $G = (V, E)$ un grafo diretto, un vertice $u \in V$ è detto *principale* se per ogni vertice $v \in V$ esiste un cammino diretto da u a v. Fornire un algoritmo in pseudo-codice che, dato un grafo diretto G, determina tutti i vertici principali di G. E' possibile che tale algoritmo abbia complessità $\mathcal{O}(|V| + |E|)$?

References

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