

Esercizi di Algoritmi e Strutture Dati

Moreno Marzolla
marzolla@cs.unibo.it

22 Marzo 2010

Cicli Dato un grafo non orientato connesso $G = (V, E)$, scrivere un algoritmo che ritorni *true* se e solo se il grafo è ciclico. Stimare la complessità asintotica dell'algoritmo proposto. Come cambia l'algoritmo si assume che il grafo G possa essere *non* connesso?

(Problema 12.8 del libro di testo) Un grafo non orientato $G = (V, E)$ si dice *bipartito* se è possibile partizionare i suoi vertici in due sottoinsiemi V_1 e V_2 tali che non esistono archi tra vertici dello stesso sottoinsieme: formalmente, per ogni $u, v \in E$, allora $u \in V_1$ e $v \in V_2$ o viceversa.

1. Dimostrare che un grafo è bipartito se e solo se non contiene cicli di lunghezza dispari, dove la lunghezza è il numero di archi nel ciclo;
2. Descrivere un algoritmo per verificare se un grafo connesso è bipartito. L'algoritmo deve esibire la partizione V_1, V_2 nel caso di risposta affermativa, e un ciclo di lunghezza dispari nel caso di risposta negativa.
3. Discutere il tempo di esecuzione dell'algoritmo proposto.

(*Suggerimento*: può essere utile considerare il fatto seguente: in un grafo bipartito, i nodi possono essere colorati con due colori in modo tale che se esiste un arco che connette i nodi u e v , allora u e v hanno colore diverso).

Componenti connesse Si consideri un grafo non orientato $G = (V, E)$. Descrivere un algoritmo che restituisca il numero di componenti connesse di G che sono alberi (ossia restituisca il numero di componenti connesse di G che non contengono cicli). (*Suggerimento*: può risultare utile considerare un algoritmo di visita di G , con opportune modifiche).