Esercizi

12. Problemi “difficili”

Esercizio 12.1
Per ognuno dei seguenti problemi di decisione, dimostrate che appartiene alla classe NP fornendo un algoritmo nondeterministico che lo risolva in tempo polinomiale.

- *Clique*
  Dato un grafo non orientato G e un intero k, stabilire se esiste un sottografo completo con k vertici.
- *Cammino hamiltoniano*
  Dato un grafo non orientato G, stabilire se esiste un cammino che visita tutti i vertici di G una sola volta.
- *Partizione*
  Dato un insieme finito A e una funzione peso p che associa a ogni elemento $a \in A$ un intero $p(a)$, stabilire se esiste un sottoinsieme $A' \subseteq A$ tale che la somma dei pesi degli elementi contenuti in A' sia uguale a quella dei pesi degli elementi non contenuti in A', cioè:
  $$\sum_{a \in A'} p(a) = \sum_{a \in A \setminus A'} p(a)\,.$$
- *Commesso viaggiatore*
  Dato un insieme finito di città $C = \{c_1, c_2, \ldots, c_m\}$ con le distanze tra esse ($d(c_i, c_j) \in \mathbb{Z}^+$ indica la distanza tra c_i e c_j) e un intero B, stabilire se esiste un “tour” di lunghezza totale al più B che visita tutte le città, cioè una sequenza $\langle c_{j_1}, c_{j_2}, \ldots, c_{j_m} \rangle$ dove $\{j_1, j_2, \ldots, j_m\} = \{1, 2, \ldots, m\}$ e
  $$\sum_{i=1}^{m-1} d(c_{j_i}, c_{j_{i+1}}) + d(c_{j_m}, c_{j_1}) \le B\,.$$
- *Colorazione di grafi*
  Dato un grafo non orientato $G = (V, E)$ e un intero k stabilire se è possibile “colorare” i vertici di G utilizzando al più k colori, in modo che i vertici adiacenti siano colorati con colori differenti.

Esercizio 12.2
Scrivete un algoritmo che risolva il problema di trovare in un grafo non orientato G il massimo numero di vertici di un sottografo completo. L’algoritmo può richiamare come procedura l’algoritmo nondeterministico che risolve il problema di decisione *Clique* dell’esercizio 12.1 (pensate una strategia che permetta di effettuare poche chiamate). Come riuscite a risolvere il problema senza utilizzare il nondeterminismo e in quanto tempo?

12 gennaio 2018

Esercizio 12.3

Scrivete un algoritmo che determini il minimo numero di colori da attribuire ai vertici di un grafo non orientato $G = (V, E)$ in modo tale che i vertici adiacenti siano colorati con colori differenti. L'algoritmo può richiamare come procedura l'algoritmo nondeterministico che risolve il problema di decisione *Colorazione di grafi* dell'esercizio 12.1 (pensate una strategia che permetta di effettuare poche chiamate).

Come riuscite a risolvere il problema senza utilizzare il nondeterminismo e in quanto tempo?

Ideate poi un algoritmo che risolva in tempo polinomiale il problema di colorare un grafo, cercando di utilizzare pochi colori, ma che non debba necessariamente determinare il numero minimo di colori. Fornite un esempio di grafo per cui l'algoritmo determina effettivamente il numero minimo di colori necessario e un esempio per cui l'algoritmo determina un numero di colori maggiore del necessario.