

Progetto di Tecniche di Programmazione

1. Implementare la classe **AdjListGraph** che implementa l'ADT Graph utilizzando la rappresentazione con liste di adiacenza.

La classe deve supportare tutti i metodi pubblici della classe Graph vista a lezione.

Implementare la funzione **load_graph**(file, directed=False) che legge da file un grafo rappresentato dalla lista dei suoi archi e costruisce (e restituisce) un oggetto della classe AdjListGraph. Nel file gli archi sono rappresentati come triple (sorgente, destinazione, elemento=1), dove elemento è inizializzato per default a 1 se non presente.

2. K-colorare un grafo significa assegnare ad ogni vertice un colore (rappresentato da un numero intero tra 1 e K) in modo tale che due vertici adiacenti non abbiano lo stesso colore. Progettare ed implementare l'algoritmo **graph_coloring**(g) che, preso in input un AdjListGraph g, restituisce un dizionario colors contenente una K-colorazione di G (colors[v] = colore assegnato al vertice v). La colorazione deve utilizzare al più D+1 colori, dove D è uguale al grado massimo dei vertici del grafo.

Analizzare la complessità dell'algoritmo proposto.

Sugg. Utilizzare la programmazione greedy.

Implementare la funzione **print_coloring**(colors, g) che, preso in input un AdjListGraph ed una sua colorazione, stampa le seguenti informazioni:

- il numero di colori utilizzati,
- il colore assegnato ad ogni vertice,
- la coppia di colori assegnati ai vertici adiacenti ad ogni arco.

3. Progettare ed implementare l'algoritmo **enumerate_coloring**(g, k) che, preso in input un AdjListGraph ed un intero k, restituisce tutte le k-colorazioni del grafo. La funzione restituisce una lista di colorazioni, ognuna rappresentata da un dizionario (vedi esercizio punto 2).

Implementare la funzione **min_colors**(g) che calcola il minor numero di colori necessari per colorare un grafo g.

La soluzione deve essere caricata sul sito del corso **almeno una settimana prima della discussione**.

Il materiale consegnato deve essere costituito da un unico file compresso contenente:

- Un file pdf con la documentazione del progetto
- Un progetto Python chiamato #gruppo_2_TdP contenente:
 - il package Python TdP_collections (con tutte le implementazioni delle strutture dati di supporto, eventualmente modificate);
 - il package Python pkg_i contenente le classi e le funzioni per la soluzione dell'esercizio i (i = 1, 2, 3, 4, 5);
 - uno script di testing per ciascun esercizio.

Nei prossimi giorni verranno caricati sul sito del corso dei data-set per testare la correttezza del progetto.