



Esercitazione di laboratorio n. 4

Esercizio n.1: Vertex Cover

Competenze: esplorazione dello spazio delle soluzioni con i modelli del Calcolo Combinatorio

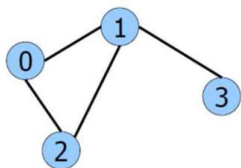
(Ricorsione e problem-solving: 3.2, 3.3)

Sia dato un grafo non orientato G di N vertici, identificati da interi nell'intervallo $0..N-1$, ed E archi, identificati come coppie di vertici. Il grafo è memorizzato su di un file, nella cui prima riga compaiono N ed E , mentre nelle E righe successive compaiono, uno per riga, gli archi nella forma $u\ v$.

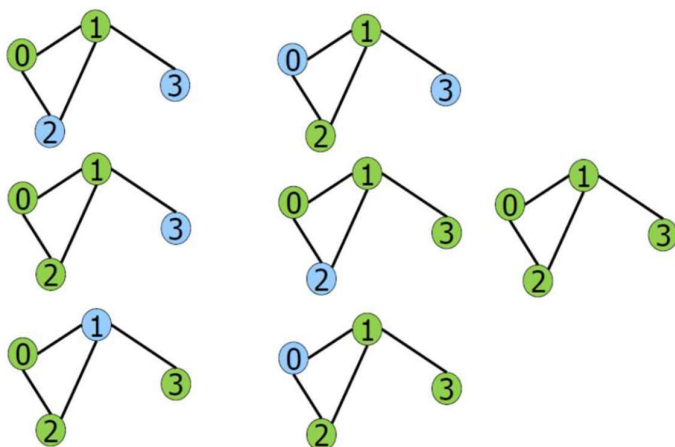
Un vertex cover è un sottoinsieme W dei vertici tale che tutti gli archi in E abbiano almeno uno dei 2 vertici su cui incidono in W : $\text{vertex_cover} = \{(u,v) \in E \text{ tali che } u \in W \text{ OR } v \in W\}$.

Dopo aver letto da file il grafo G ed aver memorizzato le informazioni rilevanti in opportune strutture dati, si elenchino tutti i vertex cover.

Esempio: per il grafo seguente



esistono i seguenti vertex cover: (0, 1), (1, 2), (0, 1,2), (0, 1,3), (0, 2,3), (1, 2,3), (0,1,2,3).



Si osservi che questo esercizio non richiede conoscenze di Teoria dei Grafi. Si identifichi il modello del Calcolo Combinatorio utilizzato.



Esercizio n.2: Playlist

Competenze: esplorazione dello spazio delle soluzioni con i modelli del Calcolo Combinatorio (Ricorsione e problem-solving: 3.2, 3.3)

Un gruppo di amici vuole preparare una playlist per un viaggio. Per accontentare tutti, ogni amico ha diritto a proporre fino a cinque canzoni tra cui scegliere. La playlist viene creata scegliendo tante canzoni quanti sono gli amici, e per ognuno di essi scegliendo una ed una sola canzone tra quelle da lui proposte. Tutte le canzoni sono distinte.

Le informazioni sulle canzoni proposte sono memorizzate in un file testuale (`brani.txt`) organizzato come segue:

- sulla prima riga appare il numero A di amici
- seguono A blocchi di righe, uno per ogni i -esimo amico, tali per cui
 - sulla prima riga del blocco appare il numero C_i di canzoni proposte dall'amico
 - seguono C_i stringhe, in ragione di una per riga, riportanti i titoli dei brani proposti

Si assuma che tutti i titoli delle canzoni non contengano spazi e siano lunghi al più 255 caratteri.

Si scriva un programma in C che, letto il file di input, generi tutte le playlist possibili con le regole di cui sopra.

Nota: i contenuti del file di input di esempio in allegato sono indentati per rendere più facile distinguere le porzioni dedicate ad ogni amico.

Si consiglia di individuare preventivamente a quale modello del Calcolo Combinatorio, tra quelli visti a lezione, faccia riferimento il problema e in seguito di adattare il codice presentato in aula.

Esercizio n.3: Collane e pietre preziose

Competenze: esplorazione dello spazio delle soluzioni con i modelli del Calcolo Combinatorio (Ricorsione e problem-solving: 3.2.4, 3.3.6), problemi di ottimizzazione (Ricorsione e problem solving: 3.5)

Un gioielliere ha a disposizione z zaffiri, r rubini, t topazi e s smeraldi per creare una collana infilando una pietra dopo l'altra. Deve però soddisfare le seguenti regole:

- uno zaffiro deve essere seguito immediatamente o da un altro zaffiro o da un rubino
- un rubino deve essere seguito immediatamente o da uno smeraldo o da un topazio
- un topazio deve essere seguito immediatamente o da uno zaffiro o da un rubino
- uno smeraldo deve essere seguito immediatamente o da un altro smeraldo o da un topazio.

Si scriva una funzione C che calcoli la lunghezza e visualizzi la composizione di una collana a lunghezza massima che rispetti le regole di cui sopra. La lunghezza della collana è il numero di pietre preziose che la compongono.

Osservazione: la lunghezza della soluzione non è nota a priori, ma può variare tra 1 e $(z+r+t+s)$.

Si identifichi il modello del Calcolo Combinatorio. Si valuti la possibilità di introdurre criteri di pruning.

Nota: si presti attenzione al crescere del tempo di esecuzione richiesto dall'identificazione della soluzione a fronte dell'aumentare dei valori dei parametri di ingresso per il problema (numero di pietre preziose disponibili).



Esercizio n.4: Collane e pietre preziose (versione 2)

Si consideri il contesto introdotto dall'esercizio precedente. Ogni tipologia di pietra è caratterizzata dal suo valore (intero non negativo) (val_z , val_s , val_r , val_t). Il valore della collana è dato dalla somma dei valori delle singole pietre che la compongono. Indicando con n_z , n_s , n_r e n_t il numero di zaffiri, smeraldi, rubini e topazi, il valore della collana è:

$$val = val_z * n_z + val_s * n_s + val_r * n_r + val_t * n_t$$

La composizione della collana deve rispettare tutte le regole introdotte nell'esercizio precedente. In aggiunta, devono essere rispettati i seguenti criteri:

- nessuna tipologia di pietra si può ripetere più di max_rip volte consecutive
- nella collana, il numero di zaffiri non può superare il numero di smeraldi

Si scriva una funzione C che calcoli la composizione di una collana a valore massimo che rispetti le regole di cui sopra.

Valutazione: Gli esercizi 1, 2 ed uno a scelta tra l'esercizio 3 e 4 saranno oggetto di valutazione
Scadenza caricamento di quanto valutato: entro le 23:59 del 30/11/2025.