

Progetto “Doppio Grafo”

Simulazione del 19 Maggio 2020

Descrizione del progetto

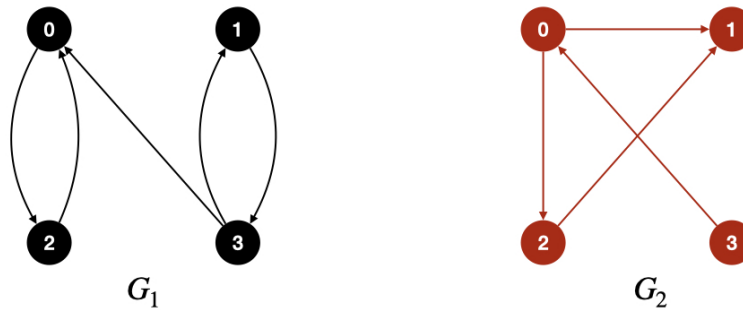
Si supponga di avere due grafi $G_1 = (V, E_1)$ e $G_2 = (V, E_2)$ sullo stesso insieme di $|V| = n$ vertici. Senza perdita di generalità supponiamo $V = \{0, 1, \dots, n - 1\}$.

Si vuole implementare un algoritmo che permetta di stabilire, per ogni coppia di nodi (i, j) , in quanti grafi esiste un percorso tra i e j .

Questi grafi vengono rappresentati come una matrice M di dimensione $n \times n$ in cui il valore in posizione (i, j) ha la seguente semantica:

- Se $M_{i,j} = 0$ (00 in binario) allora l'arco $(i, j) \notin E_1$ e $(i, j) \notin E_2$. Ovvero i nodi i e j non sono connessi da un arco in nessuno dei due grafi.
- Se $M_{i,j} = 1$ (01 in binario) allora l'arco $(i, j) \in E_1$ e $(i, j) \notin E_2$. Ovvero i nodi i e j sono connessi da un arco nel grafo G_1 ma non in G_2 .
- Se $M_{i,j} = 2$ (10 in binario) allora l'arco $(i, j) \notin E_1$ e $(i, j) \in E_2$. Ovvero i nodi i e j sono connessi da un arco nel grafo G_2 ma non in G_1 .
- Se $M_{i,j} = 3$ (11 in binario) allora l'arco $(i, j) \in E_1$ e $(i, j) \in E_2$. Ovvero i nodi i e j sono connessi da un arco in entrambi i grafi.

Una rappresentazione di due grafi di 4 vertici rappresentati in questo modo e della matrice che si vuole ottenere come risultato è presentato nella figura:



$$M = \begin{bmatrix} 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{bmatrix}$$

Numero di grafi in cui
esiste un cammino
tra i nodi i e j

$$\begin{bmatrix} 2 & 1 & 2 & 0 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 2 & 2 & 2 & 2 \end{bmatrix}$$

Codice

È richiesta l'implementazione della seguente funzione:

```
def cammini_doppio_grafo(M):  
    # restituisce una matrice della stessa dimensione di M
```

in cui l'unico argomento, la matrice M, è rappresentata come un array di array. Per esempio:

```
M = [[0, 2, 3, 0], [0, 0, 0, 1], [1, 2, 0, 0], [3, 1, 0, 0]]
```

L'output della funzione deve essere anch'esso un array di array. Nel caso d'esempio:

```
[[2, 1, 2, 0], [1, 2, 1, 1], [1, 1, 2, 0], [2, 2, 2, 2]]
```

Per l'implementazione sono possibili molteplici approcci, tra questi uno è la modifica dell'algoritmo di Floyd-Warshall.

Si raccomanda di commentare adeguatamente l'algoritmo utilizzato. Se necessario, il codice può contenere funzioni aggiuntive di supporto o di test.

Indicazioni

Il progetto deve essere svolto **individualmente**. La consegna dovrà avvenire entro le ore 23:59 del giorno xx/yy/yyyy secondo le seguenti modalità:

- Invio di una email a lmanzoni@units.it dal vostro account email istituzionale con oggetto *[informatica] consegna progetto appello del xx/yy/yyyy*.
- L'email deve avere come allegato il progetto in un singolo file in codice sorgente Python versione 3, dal nome `Nome_Cognome_matricola.py`, quindi, per esempio Mario Rossi di matricola 12345 consegnerà un file dal nome `Mario_Rossi_12345.py`.
- Il file deve contenere sotto forma di commento le seguenti linee indicanti nome, cognome e numero di matricola:

```
# Nome: Mario  
# Cognome: Rossi  
# Matricola: 12345
```