

Esercizio 1 OK

Dato un grafo diretto e pesato $G(V,E,w)$, con V rappresentante l'insieme dei nodi, E l'insieme degli archi e $w(u,v)$ indicante il peso dell'arco (u,v) , scrivere una funzione chiamata **esercizio1** che prenda in input il grafo e restituisca *true* se per ogni nodo $u \in V$, la somma dei pesi degli archi uscenti da u è sempre maggiore dei singoli pesi di tutti gli archi uscenti dai nodi adiacenti di u ; altrimenti, restituisca *false*. La condizione è sempre valida per un nodo senza archi uscenti. I nodi adiacenti di un nodo a sono tutti i nodi x tale che esiste l'arco (a,x) .

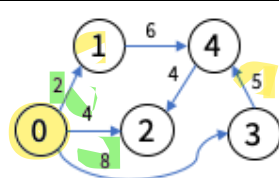
Il grafo è rappresentato da una classe *GrafoPesato*, i cui metodi di interfaccia necessari a svolgere l'esercizio sono i seguenti: data g un'istanza della classe *GrafoPesato*:

- $g.n()$ restituisce il numero di nodi del grafo,
- $g.m()$ restituisce il numero di archi del grafo,
- $g(i,j)$ restituisce un numero intero positivo p rappresentante il peso dell'arco (i,j) , se questo esiste; se l'arco (i,j) non esiste, la funzione restituisce -1.

I nodi del grafo sono etichettati da 0 a $n-1$.

Esempio: tutti i nodi soddisfano la proprietà, quindi la funzione restituisce *true*.
Ad esempio:

- per il nodo 0, somma uscenti = 14, pesi archi uscenti dai nodi adiacenti di 0: {6,5}, 14 è maggiore di entrambi;
- per il nodo 4, somma uscenti = 4, pesi archi uscenti dai nodi adiacenti di 2: 0 (nessun arco uscente), 4 è maggiore di 0.



Esercizio 2 OK

Dato in input un insieme U di stringhe, una collezione S di multiinsiemi i cui elementi appartengono ad U , e un intero k , scrivere una funzione chiamata **esercizio2** che prenda in input U , S e k , e restituisca *true* se è possibile scegliere al massimo k insiemi della collezione S tale che la loro unione sia uguale a U , altrimenti restituisca *false*. L'unione di due insiemi a e b è definita nel classico modo insiemistico (ovvero tutti gli elementi di a e b presi una sola volta). Non è possibile assumere che gli elementi in U e in S siano ordinati.

Assunzioni:

- U è rappresentato da un `vector<string>` (`ArrayList<String>` se si usa Java),
- S è rappresentato da un `vector<vector<string>>` (`ArrayList<ArrayList<String>>` se si usa Java),
- U non contiene mai la stessa stringa più volte; ogni `vector` (o `ArrayList` se si usa Java) in S può contenere la stessa stringa più volte.

Esempio: la funzione dovrà restituire *true* poiché è possibile scegliere al massimo k (ovvero 3) multiinsiemi in S tale che la loro unione sia uguale ad U .
Due soluzioni d'esempio sono:

- I due insiemi $\{xq, e, f\}$ e $\{a, b\}$, la cui unione è uguale ad U .
- I tre insiemi $\{b, e, f\}$, $\{a, b\}$ e $\{xq, f\}$, la cui unione è uguale ad U .

$U = \{ a, b, xq, e, f \}$
 $S = \{ \{xq, e, f\}, \{b, e, f\}, \{a, b\}, \{xq, f\}, \{a, xq, e\} \}$
 $k = 3$