Fondamenti di Programmazione 2 (Corso di Laurea in Informatica) Esame del 9/9/2022



Esercizio 1

Data la seguente porzione di programma, rispondere alle domande corrispondenti:

Esercizio 2

Implementare una classe Partita che contenga i seguenti campi privati string squadraCasa,squadraOspite int goalCasa, goalOspite

dove squadraCasa e squadraOspite rappresentano i nomi delle squadre che hanno disputato la partita, mentre goalCasa e goalOspite rappresentano i goal realizzati dalle corrispondenti squadre. Per questa classe, implementare il costruttore con tutti i parametri e i metodi get/set.

Implementare, successivamente, una classe GestorePartite che contenga un solo campo privato list<Partita*> partite

ed implementi i seguenti metodi:

- void aggiungiPartita(string squadraCasa, string squadraOspite, int goalCasa, int goalOspite) che aggiunga, dopo averla creata dinamicamente e usando i parametri ricevuti, una partita alla lista partite;
- string getSquadraPiuForte() const, che restituisca il nome della squadra che ha vinto più partite, tra tutte quelle (in casa e fuori casa) presenti nella lista partite;
- double mediaGoal() const, che restituisca la media dei goal realizzati nelle partite presenti nella lista *partite*;
- costruttore di copia;

Fondamenti di Programmazione 2 (Corso di Laurea in Informatica) Esame del 9/9/2022



Esercizio 3

Scrivere una funzione **esercizio3** che prenda in input un grafo orientato e pesato G, dove ogni nodo in G ha associato un valore intero positivo chiamato deposito. I pesi sugli archi sono interi strettamente maggiori di zero. La funzione deve restituire true se per ogni nodo v del grafo valgono le seguenti due condizioni:

- La somma dei pesi degli archi entranti in v è minore del suo deposito, e
- Il deposito di ogni nodo avente un arco entrante in v è maggiore o uguale al deposito di v.

Se un nodo non ha archi entranti, allora le due condizioni si assumono vere. Se almeno una di tali condizioni non è soddisfatta la funzione deve restituire *false*.

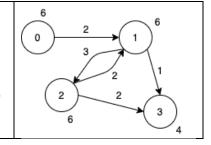
Il grafo è rappresentato da una classe GrafoPesato con la seguente interfaccia (con g un'istanza della classe):

- g.n() restituisce il numero di nodi del grafo,
- g.m() restituisce il numero di archi del grafo,
- g(i,j) restituisce il peso dell'arco tra il nodo i e il nodo j, altrimenti 0 (l'arco non esiste),
- g.dep(i) restituisce il deposito del nodo i.

I nodi sono etichettati da 0 a g.n()-1.

Esempio: in questo caso, la funzione dovrebbe restituire *true* poiché le condizioni di cui sopra valgono per ogni nodo. Ad esempio:

- il nodo 0 non ha archi entranti, quindi le proprietà si assumono vere;
- il nodo 1 ha deposito pari a 6 e due archi entranti, la cui somma è 4, quindi minore del deposito; inoltre, i nodi dei suoi archi entranti hanno tutti il deposito uguale a 6,



Esercizio 4

Scrivere una funzione **esercizio4** che prenda in input un vettore V di elementi (stringhe), un vettore T di transazioni (una transazione è vettore di stringhe v, con v appartenente a V) ed un intero positivo k. La funzione deve restituire true se esiste un sottoinsieme W di V tale che le seguenti condizioni sono soddisfatte:

- ogni coppia di elementi v_1 e v_2 in W non appare mai nella stessa transazione, e
- W ha almeno k elementi.

Se non esiste un sottoinsieme W di V tale che queste condizioni siano soddisfatte, la funzione deve restituire false.

Si può assumere che:

- Vè rappresentato da un vector<string>,
- Tè rappresentato da un vector<vector<string>>,
- Una transazione non contiene mai due stringhe uguali.

Esempio: una possibile soluzione è data dal sottoinsieme $W = \{a, e\}$. Infatti, nessuna coppia di elementi in W appare insieme in una stessa transazione, e W ha 2 elementi.

V = [a, b, c, d, e, f, g, h] T = [[a, b, f, c], [b, h, d], [b, a, c], [d, h], [e, f, g, h]] k = 2