

## ***Relazione progetto C++ - Aprile 2019***

- **Classe grafoT**

La classe grafoT è una classe templata che implementa un grafo orientato implementato tramite una matrice di adiacenza.

Essa viene implementata come una struttura dinamica dove dopo un'inizializzazione possono essere aggiunti/rimossi nodi e archi.

Per implementare la classe viene usato un puntatore ad array di nodi il quale conterrà gli identificativi dei nodi e un puntatore ad array di puntatori di booleani, ovvero la matrice di adiacenza i cui valori saranno 1 se c'è una relazione tra i nodi, 0 altrimenti.

Oltre ai metodi fondamentali, da sottolineare abbiamo:

- Funzione *init*

Questa funzione è privata e consente l'inizializzazione e la definizione di un grafo di dimensione specificata (solo inizialmente). Sostanzialmente svolge il ruolo di un costruttore primario, viene implementata separatamente per avere una migliore leggibilità del codice e anche poiché viene utilizzata anche in altri contesti (come nel copy constructor).

- Funzione *clear*

Questa funzione consente la rimozione di tutti gli elementi del grafo, implementa quello che sarebbe il lavoro del distruttore della classe.

E' stata implementata separatamente in quanto viene richiamata in tutti quei casi in cui si verifica un'eccezione per riportare la classe in uno stato coerente ed evitare memory leaks.

- Funzione *getNumNodi*

Questa funzione semplicemente ritorna il numero di nodi presenti nel grafo.

Il valore ritornato è un dato della classe che viene utilizzato al solo scopo di tenere conto di quanti nodi vengono inseriti o rimossi.

- Funzione *getNumArchi*

Questa funzione ritorna il numero di archi presenti nella matrice andando semplicemente a contare quanti 1 sono presenti in essa.

- Funzione *stampaMat*

Questa funzione stampa a video la matrice di adiacenza.

- Funzione *aggiungiNodo*

Questa funzione permette di inserire un nodo nel grafo.

Se esso è già presente: svuota il grafo e lancia un'eccezione.

Per confrontare i nodi nel grafo viene usato il funtore `_uG` che fa parte degli attributi della classe e deve essere passato dall'utente. Scelta implementativa fatta perché trattandosi di un grafo di elementi generici, deve essere l'utente a fornire un modo per confrontarli.

Questo funtore ritorna "true" se gli elementi passati in input sono uguali, "false" altrimenti.

- Funzione *rimuoviNodo*

Questa funzione consente di rimuovere un nodo dal grafo, anche in questo caso se il nodo non viene trovato si svuota il grafo e viene lanciata un'eccezione ed inoltre per gli stessi motivi di sopra viene usato il funtore `_uG` per effettuare i confronti necessari.

Se il nodo è presente nel grafo, la funzione effettua i vari aggiustamenti sulle righe e colonne della matrice.

- Funzione *aggiungiArco*

Questa funzione permette di aggiungere un arco tra 2 nodi esistenti, infatti se non viene trovato anche solamente uno dei due questa svuoterà il grafo e lancerà un'eccezione.

I nodi una volta trovati, ne viene salvata la posizione (inizialmente settata a -1) e se questa non cambia allora i nodi non sono stati trovati, altrimenti aggiunge 1 nella matrice nella posizione trovata. Le comparazioni vengono fatte sempre tramite il funtore.

- Funzione *rimuoviArco*

Questa funzione è esattamente simmetrica rispetto la precedente avendo la stessa struttura di base, con la differenza che alla fine sovrascriverà con 0 il valore nella posizione trovata nella matrice.

- Funzione exist

Questa funzione serve per verificare se un nodo è presente all'interno del grafo.

Semplicemente si itera sull'array di nodi usando il funtore `_uG` e ritorna 1 (true) o 0 (false) a seconda dell'esito.

- Funzione *hasEdge*

Questa funzione presenta una struttura simile a quella di "aggiungiArco" e "rimuoviArco", cerca la posizione dei nodi passati in input nell'array di nodi (usando sempre il funtore per fare i confronti), e se nella matrice alla posizione indicata trova un 1 allora è presente un arco tra i nodi.

- Classe *const\_iterator*

La classe `const_iterator` è interna alla classe grafo ed implementa un iteratore costante di tipo forward.

Questo iteratore scorre gli elementi dell'array che rappresenta i nodi presenti nel grafo.

- Operatore globale di stream

Viene reimplementato l'operatore di stream "`<<`" per permetterci di stampare i nodi di un grafo passato in input.

Vengono utilizzati gli iteratori implementati nella classe `const_iterator` per scorrere gli elementi nell'array di nodi.