ASD, esame di laboratorio

Grafi Molecolari

8 Gennaio 2019

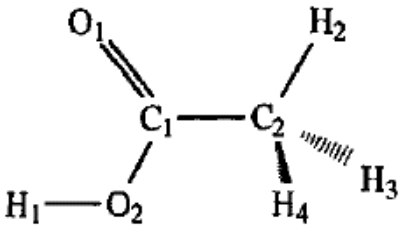
Questa prova d'esame consente di totalizzare al massimo **16** punti; poiché la prova di laboratorio contribuisce al voto finale per un massimo di 14 punti, i voti maggiori di 14 verranno riportati a 14. **Su ciascuna funzione si raggiunge il punteggio massimo indicato se la funzione è corretta e se opera con una complessità "ragionevole" e non presenta errori di stile**. Con errori di stile si intendono ad esempio l'accesso alle strutture dati che non avviene attraverso le funzioni del TDD ma in modo diretto, la mancata fattorizzazione (frammenti di codice ripetuti svariate volte invece che racchiusi in una funzione), codice poco commentato, difficile da seguire, inutilmente complesso. La soglia per raggiungere la sufficienza della prova di laboratorio è **8** punti. Il tempo a disposizione per lo svolgimento è **3 ore.**

Testo

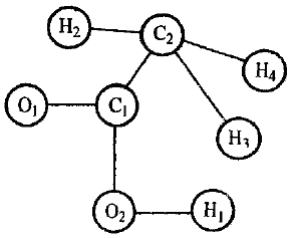
*Nella teoria chimica dei grafi, un **grafo molecolare** o **grafo chimico** è la rappresentazione della formula di struttura di un composto chimico mediante l'utilizzo di un grafo. La sua introduzione si ritiene dovuta a Arthur Cayley, il quale ne fece uso in una pubblicazione del 1874 in cui trattava gli isomeri ancora prima dell'introduzione del termine "grafo"* ... (fonte wikipedia)

In un grafo molecolare i nodi (o vertici) corrispondono agli atomi e gli archi ai legami. Esistono diversi tipi di atomi, uno per ogni elemento chimico (es. l'idrogeno abbreviato con H, l'ossigeno con O, e il carbonio con C). Ogni nodo del grafo molecolare può rappresentare uno o più atomi dello stesso tipo. Ad esempio H2 è un nodo che corrisponde a 2 atomi di idrogeno.

Consideriamo ad esempio l'acido acetico:



il suo grafo **molecolare** è il seguente:



In questo esame di laboratorio è richiesto di implementare in C++ **quattro** funzioni (specificate nell'apposita sezione mostrata sotto) che hanno come oggetto i grafi molecolari implementati con le liste di adiacenza.

La traccia che vi viene fornita contiene tre file. Il file *main.cpp* che implementa un semplice *menu* che richiama alcune operazioni del grafo molecolare e i casi di test. Il file *graph.h* che contiene i prototipi relativi alle funzioni per la creazione di un grafo molecolare e alle quattro funzioni da implementare. Il file *graph.cpp* che contiene l'implementazione delle funzioni per la creazione del grafo[1] (queste sono già fornite) e lo scheletro delle quattro funzioni da implementare.

Funzioni da implementare e relativo punteggio attribuito
Il punteggio assegnato a ciascuna funzione è indicato a fianco della funzione stessa.

```
namespace graph {
  typedef string Label;
  struct vertexNode; // definita nel file graph.cpp
  typedef vertexNode* Graph; // un grafo e' identificato dal puntatore al primo
                             nodo inserito
  const Graph emptyGraph = NULL;

  // createEmptyGraph restituisce il grafo vuoto
  Graph createEmptyGraph(); /* Già implementata */

  // Aggiunge nuovo nodo con etichetta la stringa. Fallisce se il nodo e' gia'
     presente
  bool addVertex(Label, Graph&);  /* Già implementata */

  // Aggiunge nuovo arco tra i due nodi con etichette le due stringhe .
     Fallisce se non sono presenti tutti e due i nodi o se l'arco
     tra i due e' gia' presente.
  bool addEdge(Label, Label, Graph&); /* Già implementata */

  // Restituisce true se il grafo e' vuoto, false altrimenti
  bool isEmpty(const Graph&);  /* Già implementata */

  // Verifica che tutti i nodi del grafo siano formati da una lettera MAIUSCOLA
     (semplificazione di un elemento chimico) seguito da un numero intero
     maggiore di Zero. Ritorna true in caso affermativo, false altrimenti
  bool areAllNodesNameWellFormed (const Graph&); /* Da implementare MAX punti = 4 */

  // Verifica che il grafo e' connesso
  bool connected(const Graph&);   /* Da implementare MAX punti = 4 */

  // Verifica che non esistono cappi (cappio = arco che parte e arriva sullo stesso nodo)
  bool noLoops(const Graph&);    /* Da implementare MAX punti = 4 */

  // Calcola il numero di atomi di un dato elemento (es. H) preso in input
  int computeAtomNumber(char, const Graph&); /* Da implementare MAX punti = 4 */
}
```

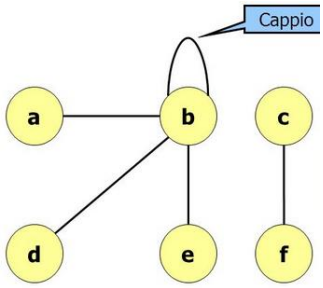
[1] L'implementazione è molto simile a quella vista a lezione anche se non identica. In particolare, in questa versione si è priviligiato l'aspetto della semplicità del codice rispetto all'information hiding

Più precisamente si richiede di implementare le seguenti funzioni :

1. **Funzione areAllNodeNamesWellFormed ()**[2]. Tale funzione verifica che i nodi del grafo siano ben formati. Ovvero verifica che tutti i nodi del grafo siano stringhe formate da una lettera MAIUSCOLA (che rappresenta una semplificazione di un elemento chimico) seguita da un numero intero maggiore di zero. Questa funzione ritorna true in caso affermativo, false altrimenti. Esempi di nodi 'ben formati' sono i seguenti: H1, C234, D10. Esempi di nodi non 'ben formati' sono i seguenti: a5, aba, Ca12, H0, H.

2. **Funzione connected().** Questa funziona verifica che il grafo inserito sia connesso**.** Un *grafo* non orientato si dice *connesso* se, per ogni coppia di vertici, esiste un cammino che li collega, altrimenti si dice sconnesso. Il grafo molecolare mostrato nella prima pagina è connesso. Se ad esempio eliminiamo l'arco C1-O2 allora otteniamo un grafo sconnesso.

3. **Funzione noLoops().** Questa funzione verifica che nel grafo non esistano cappi. Un arco che ha due estremi coincidenti si dice *cappio.*



4. **Funzione computeAtomNumber().** Tale funzione è in grado di calcolare il numero di atomi di un certo elemento chimico presente in un grafo molecolare. Ad esempio se l'utente decide di calcolare il numero di atomi totali di idrogeno (simbolo H) rispetto al grafo molecolare mostrato nella prima pagina, il software dovrà stampare a video il valore 10 (2+4+3+1). Invece se l'input è ossigeno (simbolo O) l'output dovrà essere 3.

Funzioni di Test

Il main include alcune funzioni di testing che corrispondonio alle opzioni 7, 8, 9, 10 del menù. L'utilizzo di queste opzioni è consigliato durante la prova di esame ma facoltativo.

E' importante sottolineare che il testing che mettiamo a disposizione NON E' IN ALCUN MODO ESAUSTIVO. Il testing non è mai esaustivo e per sua natura non dà certezze sulla correttezza del programma. E' opportuno effettuare altri test in particolar modo sui casi limite.

Inserimento di un grafo molecolare

Avviene selezionando l'opzione 1 del menù. Successivamente occorre inserire gli archi del grafo molecolare seprando i nodi con uno spazio e andando a capo. Il carattere '0' (zero) segna la fine

[2] Un'altra opzione sicuramente migliore di quella proposta nel testo di esame sarebbe stata quella di non permettere l'inserimento nel grafo di nodi 'mal formati'. Per scopi didattici invece è permesso l'inserimento di qualsiasi stringa, postponendo la verifica in un secondo tempo.

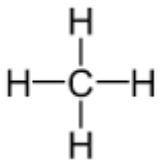
dell'input. Per inserire in memoria il grafo dell'esempio mostrato sopra occorrerà inserire i seguenti archi (non necessariamente con lo stesso ordine) e terminare con uno '0':

```
Fornisci la tua scelta ---> 1
------------------------------------------------------------------------

Inserisci gli archi del grafo molecolare: origine destinazione [invio] (0 per terminare)
H2 C2
C2 H4
C2 H3
C2 C1
C1 O1
C1 O2
O2 H1
0
```

Suggerimenti, chiarimenti e vincoli:

- Per semplicità si suppone che l'elemento chimico sia sempre composto da un solo carattere (ad esempio H per l'idrogeno). Quindi con il nostro software non sarà possibile gestire elementi chimici formati da due o più caratteri (ad esempio il calcio, Ca).
- Sempre per semplicita', si trascura di verificare che i nomi degli atomi corrispondano a elementi riportati nella tavola periodica (per esempio, è consentito avere un atomo etichettato "Q", che non corrisponde ad alcun elemento chimico).
- Il numero di atomi di un elemento chimico è un intero non necessariamente compreso tra 1 e 9. Ad esempio H12 indica 12 atomi di idrogeno.
- Per semplicità nel grafo molecolare non possono esserci due nodi aventi la stessa label. Ad esempio il nostro software non potrà gestire il seguente grafo molecolare.



- Possiamo assumere che le funzione computeAtomNumber() sia sempre chiamata su un grafo molecolare avente tutti i nodi 'ben formati'. Ovvero solo se la funzione areAllNodesNameWellFormed () ritorna true su tale grafo.
- E' possibile modificare il file graph.cpp ma non i file main.cpp e graph.h. Se occorre è ammesso aggiungere file al progetto.
- Non è permesso usare la libreria STL .
- Nel file graph.cpp il tipo di dato grafo è stato implementato con le liste di adiacenza nella variante non ottimizzata[3].
- Nello svolgimento della prova di esame potrebbe essere utile consultare la tabella ASCII. E' possibile consultare la tabella utilizzando il comando "man ascii" dalla shell di Linux.

[3] L'implementazione proposta differisce leggermente da quella vista a lezione