Versace Alessandro s246056

Relazione problema 18 punti appello 28/01/2020

Per la risoluzione ho adottato un approccio top-down implementando prima il main e richiamando le relative funzioni che poi sono andato ad implementare adattandole alle richieste specifiche.

Siccome il file di input era in formato standard ho utilizzato la funzione dei grafi standard "GRAPHload" per l'acquisizione. Alla struttura del grafo ho aggiunto una matrice di interi "eliminati" per registrare con degli 1 gli archi eliminati, utile poi per il terzo punto. Questa matrice è stata aggiunta all'allocazione e free come estensione delle normali funzioni già fornite nel "Graph.h".
Ho successivamente deciso di implementare esternamente nel main un vettore di interi "degree" di dimensione uguale al numero di vertici, che ho usato per registrare le informazioni dei gradi dei vari vertici (utilizzando la corrispondenza intero-vertice tramite la tabella di simboli già presente nel grafo), ricavandoli attraversando la matrice delle adiacenze con l'apposita funzione creata "GRAPHcalculateDegree".

Per la seconda richiesta (k-core) ho creato una funzione apposita "GRAPHk_core", che iterativamente va a controllare se ci sono vertici con un degree minore della k acquisita da tastiera. La funzione fa uso del vettore precedentemente creato "degree" andando a mettere a -1 i vertici con un degree minore e diminuendo il degree di tutti i vertici a questo collegati. Iterativamente prosegue fino a quando fa un ciclo completo sul vettore dei gradi senza apportare modifiche.

La terza richiesta viene implementata utilizzando una funzione di combinazioni semplici sugli archi del grafo (non è importante l'ordine e non vanno ripetuti archi), per creare un sottoinsieme di archi di dimensione j, enumerando tutti i casi. Nella condizione terminale si fa un controllo sulle componenti connesse dopo la rimozione di questi archi scelti, tramite la funzione standard "GRAPHcc" per le componenti connesse leggermente modificata aggiungendo il controllo per evitare gli archi rimossi.
Questa termina nel momento in cui trova una soluzione valida, cioè quando rileva più di una componente connessa.

Rispetto alla versione scritta in aula:

- Implementate tutte le funzioni di libreria
- Implementate le funzioni dal main meno importanti, saltate durante lo scritto
- Implementate leggere modifiche alle funzioni standard di libreria Graph.h su init, free e GRAPHcc per ma matrice "eliminati"
- Nella versione in classe riducevo a metà i gradi dopo averli calcolati nel vettore degree, sbagliando
- Non avevo implementato un controllo per il terzo punto per verificare che il grafo fosse j-edge-connected
- Modifiche all'implementazione delle combinazioni semplici, facendo uso di un vettore di archi da disporre creato tramite GRAPHedges, invece di utilizzare la strategia di esplorazione come nella versione scritta.