

Niccolò Dimonte
Matricola s248547

Prova di programmazione da 18 punti (31/01/19)

Struttura Dati: la struttura dati utilizzata è un ADT I classe per il Grafo e per la lista Edge. Per la lista di vertici dal secondo file ho utilizzato un vettore sovrallocato dinamicamente di struct, con eventuale reallocazione nel caso serva dimensione maggiore.

Scelte Algoritmiche: le principali operazioni sono

- inizializzazione grafo;
- memorizzazione dati da file;
- scorrimento liste per funzioni.

Acquisiamo da riga di comando i nomi dei file e quindi leggiamo il grafo dal primo file, acquisendo i dati degli archi in una lista Edge grazie alle funzioni GraphLoad e EdgeCreate. Dopodichè passiamo ad acquisire i dati dal secondo file tramite la funzione VertexLoad, che all'interno sovralloca dinamicamente la memoria del vettore di tipo Vertex.

Passiamo il nostro Grafo alla funzione FileKernel per fare il controllo sulla lista di vertici del secondo file, precedentemente salvata in Vertex. All'interno di FileKernel ci sono due funzioni di controllo (Check1 e Check2) che ritornano 1 e 0 in base alle condizioni necessarie per far sì che un nodo k sia nell'insieme del kernel. In base al valore ritornato, la funzione FlagResult stampa il Kernel trovato.

Note: mi sono attenuto al programma cartaceo scritto in classe, sul quale son riuscito a scrivere solo i primi tre punti del programma. Le strutture dati potevano risultare più efficienti grazie all'uso di una SymbolTable.